

TUGAS AKHIR - KI141502

Deteksi Pergerakan Tangan dari Data EEG Menggunakan Fast Walsh-Hadamard Transform dan Artificial Neural Network

Nurhamidah Tyas Palupi
NRP 05111440000057

Dosen Pembimbing I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

Deteksi Pergerakan Tangan dari Data EEG Menggunakan Fast Walsh-Hadamard Transform dan Artificial Neural Network

Nurhamidah Tyas Palupi
NRP 05111440000057

Dosen Pembimbing I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2018

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

HAND MOTION DETECTION FROM EEG SIGNAL DATASET USING FAST WALSH-HADAMARD TRANSFORM AND ARTIFICIAL NEURAL NETWORK

Nurhamidah Tyas Palupi
NRP 05111440000057

Supervisor I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom

Supervisor II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty Of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya, 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Deteksi Pergerakan Tangan dari Data EEG Menggunakan *Fast Walsh–Hadamard Transform* dan *Artificial Neural Network*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :
Nurhamidah Tyas Palupi
NRP : 05111440000057

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
NIP. 19490823 197603 2 001

(pembimbing 1)

Dini Adni Navastara, S.Kom., M.Sc.
NIP. 19851017 201504 2 001

(pembimbing 2)

SURABAYA
JUNI 2018

[Halaman ini sengaja dikosongkan]

DETEKSI PERGERAKAN TANGAN DARI DATA EEG MENGGUNAKAN *FAST WALSH–HADAMARD TRANSFORM* DAN *ARTIFICIAL NEURAL NETWORK*

Nama Mahasiswa : Nurhamidah Tyas Palupi
NRP : 05111440000057
Jurusan : Departemen Informatika FTIK-ITS
Dosen Pembimbing 1 : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

ABSTRAK

Brain computer interface merupakan sebuah teknologi yang memungkinkan individu dapat berkomunikasi dengan mesin tanpa melibatkan adanya pergerakan otot. Di dalam aplikasi BCI terbaru terdapat beberapa metode, diantaranya seperti *electroencephalography* (EEG), *electrocorticography* (EcoG), dan *functional magnetic resonance imaging* (fMRI). Namun dari ketiga metode tersebut EEG adalah metode yang lebih banyak digunakan, hal ini karena kemudahan dalam mendapatkan data serta menerapkannya.

Dalam tugas akhir ini, penulis menerapkan sebuah metode untuk mendeteksi pergerakan tangan kanan atau kiri dari data sinyal EEG dengan menggunakan transformasi fitur *Fast Walsh-Hadamart Transform* (FWHT) dan metode klasifikasi *Artificial Neural Network* (ANN). Berdasarkan hasil pengujian yang dilakukan, menggunakan dua *hidden layer* dengan jumlah neuron [5 5] dan nilai *learning rate* 10 didapatkan akurasi tertinggi yaitu 73,79%.

Kata kunci: *Brain computer interface (BCI)*, *electroencephalography (EEG)*, *Fast Walsh-Hadamart Transform*, *Artificial Neural Network (ANN)*.

[Halaman ini sengaja dikosongkan]

HAND MOTION DETECTION FROM EEG SIGNAL DATASET USING FAST WALSH-HADAMARD TRANSFORM AND ARTIFICIAL NEURAL NETWORK

Student Name : Nurhamidah Tyas Palupi
Student ID : 05111440000057
Major : Informatics Department FTIK-ITS
1st Supervisor : Dr.Eng. Chastine Fatichah, S.Kom., M.Kom
2nd Supervisor : Dini Adni Navastara, S.Kom., M.Sc.

ABSTRACT

Brain computer interface is a technology that allows individuals to communicate with machines without involving muscle movement. In the latest BCI applications there are several methods, such as electroencephalography (EEG), electrocorticography (EcoG), and functional magnetic resonance imaging (fMRI). But from the three methods above EEG is one of it which widely used, this because of the ease in obtaining and applying the data.

In this final project the author implements a method for detecting left or right hand movements of EEG signals by using Fast Walsh-Hadamart Transform (FWHT) for feature extraction and Artificial Neural Network (ANN) for classification. Based on the results of tests performed, using two hidden layers with number of neurons [5 5] and learning rate 10 it can obtaine the highest accuracy that is 73,79%.

Keywords: Brain computer interface (BCI), electroencephalography (EEG), Fast Walsh-Hadamart Transform, Artificial Neural Network (ANN).

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji syukur bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**DETEKSI PERGERAKAN TANGAN DARI DATA EEG MENGGUNAKAN FAST WALSH-HADAMARD TRANSFORM DAN ARTIFICIAL NEURAL NETWORK (ANN)**” dan berhasil menyelesaikan studi di Informatika ITS tepat waktu.

Pengerjaan tugas akhir ini menjadi suatu pengalaman yang tidak terlupakan bagi penulis. Untuk itu pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom dan Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing penulis yang telah memberi nasihat, arahan dan juga semangat sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku kepala Laboratorium KCV yang bersedia meminjamkan salah satu komputer untuk pengerjaan tugas akhir.
3. Orang tua penulis, Mas Bayu selaku kakak penulis dan dek Indah selaku adik penulis yang senantiasa memberikan dukungan dan semangat, serta doa-doa yang tak pernah berhenti dipanjatkan demi kesuksesan penulis.
4. Ade, Afiif, Datin, Dini, Lucha, Mala, Sita, Tionia, Upik selaku teman penulis yang selalu menemani, membantu dan memberi semangat kepada penulis.
5. Teman-teman *user* TA KCV dan *admin* KCV yang membantu dan menemani pengerjaan tugas akhir penulis.
6. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis

mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2018

Nurhamidah Tyas Palupi

DAFTAR ISI

LEMBAR PENGESAHAN.....	Kesalahan! Bookmark tidak ditentukan.
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
DAFTAR KODE SUMBER	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Tugas Akhir	2
1.5 Manfaat Tugas Akhir	3
1.6 Metodologi.....	3
1.7 Sistematika Laporan.....	4
BAB II DASAR TEORI.....	7
2.1 <i>Brain Computer Interface</i>	7
2.2 <i>Electroencephalography</i> (EEG).....	8
2.3 Transformasi fitur	Kesalahan! Bookmark tidak ditentukan.
2.4 Fast Walsh-Hadamard Transform.....	11
2.5 Artificial Neural Network	12
2.6 Metode Evaluasi Kinerja.....	14
BAB III PERANCANGAN.....	16
3.1 Perancangan Sistem	17
3.2 Perancangan Data.....	18
3.3 Perancangan Proses.....	20
3.3.1 <i>Pre-processing</i>	20
3.3.2 Ekstraksi Fitur	23
3.3.3 Klasifikasi	25

BAB IV IMPLEMENTASI.....	27
4.1 Lingkungan Implementasi.....	27
4.2 Implementasi Program	27
4.2.1 Implementasi Tahap <i>Preprocessing</i>	27
4.2.2 Implementasi Tahap Transformasi fitur.....	28
4.2.3 Implementasi Klasifikasi.....	30
BAB V UJI COBA DAN EVALUASI.....	34
5.1 Lingkungan Uji Coba.....	35
5.2 Data Uji Coba.....	35
5.3 Skenario Uji Coba.....	36
5.3.1 Skenario Uji Coba 1	36
5.3.2 Skenario Uji Coba 2	37
5.3.3 Skenario Uji Coba 3	39
5.3.4 Skenario Uji Coba 4	40
5.3.5 Skenario Uji Coba 5	41
5.4 Evaluasi Hasil Uji Coba.....	42
BAB VI KESIMPULAN DAN SARAN	45
6.1 Kesimpulan	45
6.2 Saran	45
DAFTAR PUSTAKA	47
LAMPIRAN	49
BIODATA PENULIS	75

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Pemasangan Elektroda Pada Teknik 10-20	9
Gambar 2.2 Metode FWHT	11
Gambar 2.3 Contoh Walsh Hadamard Transform	12
Gambar 2.4 Contoh Fast Walsh Hadamard Transform	12
Gambar 2.5 Visualisasi proses ANN [10]	13
Gambar 3.1 Diagram Alir Sistem	17
Gambar 3.2 Contoh Data Sinyal C3	18
Gambar 3.3 Contoh Data Sinyal Cz	19
Gambar 3.4 Contoh Data Sinyal C4	19
Gambar 3.5 Diagram Alir <i>Preprocessing</i>	20
Gambar 3.6 Contoh Input Sinyal C3	21
Gambar 3.7 Contoh Input Sinyal C4	21
Gambar 3.8 Contoh Input Sinyal C3 Baru	22
Gambar 3.9 Contoh Input Sinyal C4 Baru	22
Gambar 3.10 Hasil FWHT Channel C3	24
Gambar 3.11 Hasil FWHT Channel C4	24
Gambar 3.12 Diagram Alir Transformasi fitur	25
Gambar 3.13 Diagram Alir Klasifikasi	26

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 5.1 Hasil Akurasi Skenario Uji Coba 1.....	37
Tabel 5.2 Hasil Akurasi Skenario Uji Coba 2.....	38
Tabel 5.3 Hasil Akurasi Skenario Uji Coba 3.....	39
Tabel 5.4 Hasil Akurasi Skenario Uji Coba 4.....	40
Tabel 5.5 Hasil Akurasi Skenario Uji Coba 5.....	41

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Tahap <i>Preprocessing</i>	28
Kode Sumber 4.2 Fungsi Pemanggilan FWHT	28
Kode Sumber 4.3 Implementasi Tahap Transformasi fitur	29
Kode Sumber 4.4 Pengambilan fitur untuk klasifikasi	30
Kode Sumber 4.5 Penggabungan Fitur untuk Tahap Klasifikasi	31
Kode Sumber 4.6 Fungsi untuk <i>Kfold Cross Validation</i>	31
Kode Sumber 4.7 Implementasi Untuk Tahap Klasifikasi Menggunakan ANN	32
Kode Sumber 4.8 Implementasi Klasifikasi Menggunakan Metode <i>Naive Bayes</i> , <i>SVM</i> , dan <i>Random Forest</i>	32

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir.

1.1 Latar Belakang

Brain computer interface (BCI) adalah sistem teknologi yang memungkinkan orang berkomunikasi dengan mesin tanpa melibatkan adanya gerak otot. Pada dasarnya, BCI terdiri dari perolehan sinyal, penggalian fitur dan tahap klasifikasi. Metode seperti *electroencephalography* (EEG), *electrocorticography* (ECoG), *functional magnetic resonance imaging* (fMRI) digunakan dalam aplikasi terbaru BCI. Tetapi pada umumnya, EEG lebih banyak digunakan karena kemudahan dalam mendapatkan dan menerapkan datanya [4].

Kebanyakan studi tentang EEG dimaksudkan untuk memperbaiki keseluruhan kinerja yang telah ada sebelumnya. Untuk bisa mendapatkan akurasi yang tinggi dalam mengolah data EEG diperlukan metode transformasi fitur dan juga metode klasifikasi yang tepat. Tetapi di sisi lain, belum ada metode transformasi fitur yang secara umum bisa memberikan hasil yang maksimal. Oleh karena itu banyak sekali peneliti mengembangkan metode transformasi fitur yang berbeda dan baru demi mendapatkan hasil yang lebih maksimal. Dari beberapa literatur yang telah ada, berbagai jenis transformasi fitur telah digunakan untuk mengkarakterisasi sinyal EEG seperti *autoregressive* model [1], *common spatial pattern* [2], *Fast Walsh Hadamard Transform* [4] dan *power spectral density* [5].

Pada tugas akhir ini penulis akan melakukan deteksi pergerakan tangan dari data sinyal EEG menggunakan *Fast Walsh–Hadamard Transform* sebagai metode transformasi fitur, dan *Artificial Neural Network* sebagai metode klasifikasi. Diharapkan

metode ini dapat menghasilkan performa yang baik dalam deteksi pergerakan tangan dari data EEG.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana menerapkan transformasi fitur *Fast Walsh–Hadamard Transform* untuk mengklasifikasikan data sinyal EEG?
2. Bagaimana mendeteksi pergerakan tangan dari data EEG menggunakan *Artificial Neural Network*?
3. Bagaimana mengukur tingkat akurasi metode yang digunakan?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain sebagai berikut:

1. Dataset yang digunakan adalah Data Set III dari *BCI Competition 2003*.
2. Kelas pergerakan tangan dibagi menjadi 2, yaitu pergerakan tangan kanan dan pergerakan tangan kiri.
3. Metode ini diimplementasikan menggunakan MATLAB 2017a.

1.4 Tujuan Tugas Akhir

Tujuan dari tugas akhir ini adalah untuk mendeteksi pergerakan tangan kanan/kiri melalui sinyal EEG dari Data Set III, *BCI Competition 2003* menggunakan transformasi fitur *Fast Walsh–Hadamard Transform*.

1.5 Manfaat Tugas Akhir

Tugas akhir ini diharapkan bisa menambah wawasan tentang sinyal EEG dan juga dapat dijadikan sebagai referensi untuk penelitian tentang sinyal EEG di masa yang akan datang.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini yaitu:

1. Studi literatur

Pada tahap ini dilakukan pencarian literatur berupa jurnal atau paper yang digunakan sebagai referensi untuk pengerjaan tugas akhir ini. Literatur yang dipelajari pada pengerjaan Tugas Akhir ini berasal dari jurnal ilmiah yang diambil dari berbagai sumber di internet, beserta berbagai literatur online tambahan terkait *Artificial Neural Network*, dan *Fast Walsh–Hadamard Transform*. Makalah yang digunakan sebagai acuan adalah “*Classification of EEG Signals Recorded During Right/Left Hand Movement Imagery Using Fast Walsh Hadamard Transform Based Features*” [4] dan “*Feature selection for neutral vector in EEG signal classification*” [3].

2. Analisis dan Desain Perangkat Lunak

Untuk membangun aplikasi deteksi pergerakan tangan dari data sinyal EEG ini harus melalui beberapa tahap yaitu resizing data rekam sinyal EEG kemudian melakukan transformasi fitur. Dari proses transformasi fitur ini akan didapatkan fitur-fitur baru yang bisa digunakan untuk proses klasifikasi. Melalui proses klasifikasi inilah akan didapatkan hasil yang selanjutnya bisa digunakan untuk mendeteksi pergerakan tangan.

3. Implementasi Perangkat Lunak

Pada tahap ini akan dilaksanakan implementasi metode dan algoritma yang telah direncanakan. Implementasi perangkat lunak dilakukan di dalam platform desktop dengan menggunakan bahasa pemrograman

MATLAB menggunakan kakas bantu IDE MATLAB 9.2 (R2017a) pada platform *desktop*.

4. Pengujian dan evaluasi

Pengujian akurasi sistem yang dihasilkan dari tugas akhir ini menggunakan metode *10-fold cross validation*. Dataset akan dibagi menjadi 10 *fold*, kemudian pada sistem akan dilakukan proses training dan testing sebanyak 10 kali iterasi. Dalam setiap iterasi ke-*i*, *fold* ke-*i* akan digunakan sebagai *testing set*, sedangkan (*k*-1) *fold* sisanya akan dipakai sebagai *training set*. Proses ini akan dijalankan sampai semua *fold* terpakai sebagai *testing set*, dan hasil akurasi setiap iterasi akan dicatat, serta dicari rata-ratanya untuk menemukan hasil akurasi rata-rata dari sistem yang dihasilkan dalam tugas akhir ini.

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Perancangan

Bab ini berisi tentang analisis dan perancangan desain sistem segmentasi area *trabecular bone*.

Bab IV Implementasi Sistem

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

Bab V Uji Coba dan Evaluasi

Bab ini membahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk kinerja dari aplikasi yang dibangun.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi ke depannya.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai *dental panoramic radiograph*, *trabecular bone* dan *Region of Interest* (ROI) sebagai ruang lingkup dan studi kasus, metode *Gabor filter* untuk transformasi fitur, *K-means clustering* dalam proses *clustering*, metode *Root-Guided Decision Tree* untuk klasifikasi, serta morfologi untuk memperbaiki segmen citra biner.

2.1 *Brain Computer Interface*

Brain Computer Interface (BCI) adalah sebuah sistem yang menyediakan jalur komunikasi antara otak manusia dengan mesin/komputer. Jalur komunikasi yang dimaksud disini berbeda dengan jalur komunikasi yang semestinya terjadi antara otak sebagai sistem saraf pusat, dengan organ-organ lain pada sistem saraf peripheral tubuh [14]. Jika bisaanya komunikasi ini melibatkan koordinasi sel-sel saraf dan otot, BCI menyediakan suatu jalur komunikasi baru, dimana tidak ada keterlibatan saraf dan otot motorik tubuh, melainkan melalui sebuah mekanisme yang melibatkan biosignal.

Biosignal merupakan fitur khusus yang berasal dari tubuh manusia, sebagai suatu mekanisme reaksi dari aktivitas-aktivitas biologis yang terjadi dalam tubuh [15]. *Biosignal* untuk aktivitas otak yang digunakan sebagai masukan dari sistem BCI berasal dari aktivitas perekaman seperti *Electroencephalography* (EEG), *electrocorticography* (ECoG), dan *functional magnetic resonance imaging* (fMRI). Sebagai masukan untuk BCI, aktivitas ini dapat diperoleh melalui sebuah alat yang berfungsi secara khusus untuk merekam aktivitas otak. Sinyal yang terekam ini kemudian dapat diekstrak dan diterjemahkan oleh sistem BCI, dengan melibatkan algoritma-algoritma tertentu.

Dalam beberapa tahun terakhir, riset-riset seputar BCI telah berkembang dengan pesat. Salah satu manfaat yang diperoleh dari kemajuan pengembangan BCI adalah menyediakan sarana komunikasi untuk pasien dengan gangguan saraf motorik. Tidak dilibatkannya saraf otot motorik sebagai fitur dari sistem BCI menyebabkan intensi yang dikeluarkan otak, dalam hal ini biosignal, dapat diterjemahkan secara langsung tanpa memerlukan respon gerakan motorik yang dimaksud. Salah satu jenis biosignal yang dapat diperoleh dari aktivitas otak ini adalah sinyal *electroencephalography* yang akan dipaparkan lebih jelas pada subbab 2.2.

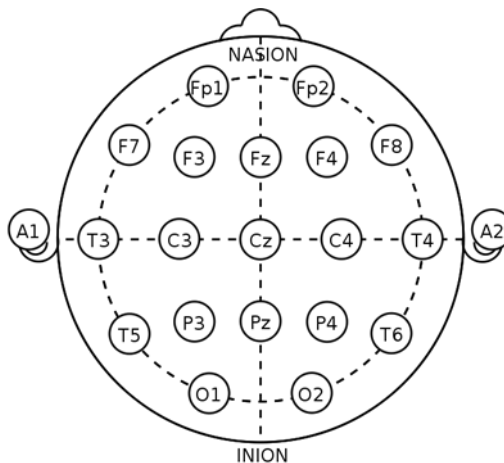
2.2 *Electroencephalography* (EEG)

Elektroencephalography/Elektro Enselo Grafi (EEG) adalah suatu alat yang mempelajari gambar dari rekaman aktifitas listrik di otak, termasuk teknik perekaman EEG dan interpretasinya. EEG mengukur fluktuasi tegangan yang dihasilkan oleh arus ion di dalam neuron otak [6]. Neuron-neuron di korteks otak mengeluarkan gelombang-gelombang listrik dengan voltase yang sangat kecil (mV), yang kemudian dialirkan ke mesin EEG untuk diamplifikasi sehingga terekamlah elektroenselogram yang ukurannya cukup untuk dapat ditangkap oleh mata pembaca EEG sebagai gelombang alfa, beta, theta dan sebagainya.

Transformasi sinyal EEG menjadi suatu model, merupakan suatu cara yang sangat efektif dalam membantu klasifikasi sinyal EEG, mengidentifikasi serta mengestimasi spektrum sinyal EEG. Sinyal EEG mengandung komponen-komponen tertentu, yang dikenal sebagai gelombang alfa (8-13 Hz), beta (14-30 Hz), teta (4-7 Hz), dan delta (0.5-3 Hz), sehingga transformasi sinyal EEG menjadi daerah-daerah frekuensi merupakan hal yang sangat berguna, terutama dalam identifikasi gelombang-gelombang di otak [9].

Salah satu teknik *Electroencephalography* yang paling populer sekaligus dijadikan standardisasi secara internasional adalah *The International 10-20 System*. Teknik ini memiliki fungsi

utama untuk mendeskripsikan atau melakukan pemasangan elektroda-elektroda pada kepala manusia, dalam sebuah penelitian atau eksperimen yang menyangkut BCI maupun EEG. Tujuan adanya standardisasi dengan sistem 10-20 ini adalah agar ada standar yang sama, yang dapat diterapkan pula pada penelitian-penelitian selanjutnya. Hal ini akan memungkinkan subyek serta metode yang digunakan dapat diperbandingkan dengan penelitian serumpun dari waktu ke waktu, terlepas dari perbedaan karakteristik maupun algoritma translasi yang digunakan. Nilai 10-20 pada teknik ini merujuk pada fakta dimana pemasangan elektroda-elektroda dilakukan sedemikian rupa hingga jarak masing-masingnya adalah sekitar 10-20% dari jarak keseluruhan elektroda. **Gambar 2.1** menunjukkan ilustrasi implementasi teknik ini.



Gambar 2.1 Ilustrasi Pemasangan Elektroda Pada Teknik 10-20 [14]

Dalam Gambar 2.1, setiap huruf melambangkan sebuah lokasi khusus pemasangan elektroda pada daerah kepala. Huruf F melambangkan daerah frontal, T melambangkan temporal, C melambangkan central, P untuk daerah parietal, dan O untuk occipital. Angka genap seperti 2, 4, 6, dan 8 digunakan pada daerah

sebelah kanan kepala, sedangkan angka ganjil seperti 1, 3, 5, dan 7 untuk daerah sebelah kiri. Konsonan z, yang berarti zero, merujuk pada elektroda yang terletak di garis tengah. Huruf A yang pada kedua ujung sistem merujuk pada daerah telinga.

Pada tugas akhir ini, dataset sinyal rekaman EEG yang digunakan adalah Data Set III dari *BCI Competition 2003*. Dataset ini berisi set sinyal EEG yang didapat dari seorang subyek normal (wanita, 25 tahun). Subyek duduk di sebuah kursi dengan keadaan tangan santai. Kemudian subyek diminta untuk membayangkan sedang menggerakkan tangannya, order untuk menggerakkan tangan kanan/kiri dilakukan secara acak. Hasil sinyal EEG yang diperoleh kemudian diproses melalui sistem yang disusun, sehingga dapat menghasilkan keluaran berupa kelas gerakan yang dimaksud.

2.3 Transformasi fitur

Transformasi fitur adalah suatu proses untuk mengolah data yang ada, hingga menghasilkan set fitur baru yang meminimalisir redundansi antar fitur. Data sinyal EEG yang didapat seringkali berdimensi besar, baik itu terdiri dari banyak channel elektroda, ataupun memiliki frekuensi yang tinggi. Hal ini menyebabkan proses klasifikasi pada data nantinya akan berjalan tidak efektif, karena waktu pemrosesan akan sangat Panjang. Terkait dengan besarnya dimensi, hasil akurasi pun akan rendah akibat adanya percampuran nilai informasi antar fitur. Adalah fungsi metode pada transformasi fitur, untuk mereduksi dimensi yang besar dan tidak efisien pada data yang ada, sehingga menghasilkan fitur hasil ekstraksi yang tetap mengandung nilai informasi yang tinggi dari data mentah, namun dengan jumlah dimensi yang lebih kecil dan efektif ketika akan diklasifikasi. Dalam tugas akhir ini metode transformasi fitur yang digunakan adalah *Fast Walsh-Hadamard Transform* (FWHT).

2.4 Fast Walsh-Hadamard Transform

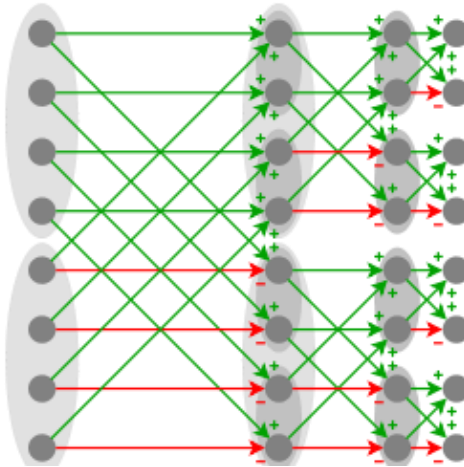
Fast Walsh Hadamard Transform (FWHT) merupakan metode pengolahan sinyal dimana dapat diartikan sebagai versi general dari *Fourier Transform*. FWHT adalah transformasi orthogonal dan *lossless*. Fungsi dasar dari transformasi ini adalah fungsi *Walsh Hadamard Transform* (WHT) yang mengekstraksi informasi waktu dan frekuensi hanya memiliki dua nilai, +1 dan/atau -1 [11].

Hasil dari $2^n \times 2^n$ WHT matrik adalah H_n . Persamaan untuk matriks H_n dapat dilihat dari persamaan 2.1 berikut:

$$H_n = \begin{pmatrix} H_{n-1} & \cdots & H_{n-1} \\ \vdots & \ddots & \vdots \\ H_{n-1} & \cdots & -H_{n-1} \end{pmatrix} \quad 2.1$$

$$Y = H_n \cdot X \quad 2.2$$

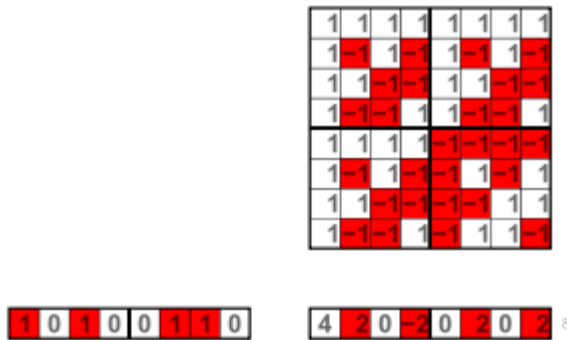
Dimana $n=1, 2, 3 \dots$ [12]. *Output* sinyal (Y) didapatkan dari perkalian antara *input* sinyal (X) dan H_n . Persamaan Y dapat dilihat pada persamaan 2.2.



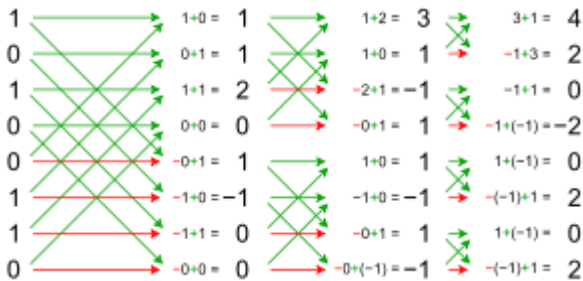
Gambar 2.2 Metode FWHT [10]

Sedangkan dengan metode FWHT penghitungan dilakukan dengan menggunakan algoritma *divide and conquer* yang secara rekursif memecah WHT ukuran n menjadi dua WHT ukuran yang lebih kecil $n/2$ [13]. Untuk lebih jelasnya bisa dilihat pada **Gambar 2.2**.

Perbandingan perhitungan WHT dengan FWHT bisa dilihat dari contoh **Gambar 2.3** dan **Gambar 2.4**.



Gambar 2.3 Contoh Walsh Hadamard Transform [15]

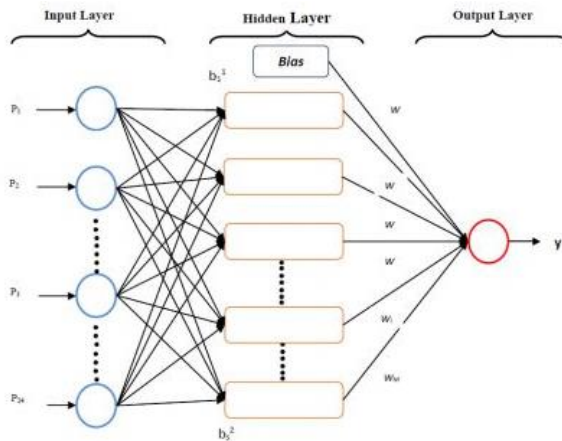


Gambar 2.4 Contoh Fast Walsh Hadamard Transform [10]

2.5 Artificial Neural Network

Artificial neural network (ANN) terinspirasi dari kesadaran atas *complex learning system* pada otak yang terdiri dari set set neuron yang saling berhubungan secara dekat. Jaringan

neuron mampu melakukan tugas yang sangat kompleks seperti klasifikasi dan pemahaman pola. ANN dapat memperkirakan rentang yang cukup luas suatu model statistika dan fleksibel dalam menggambarkan model (*linier* maupun *nonlinier*) [8]. ANN dapat digunakan untuk permasalahan yang sama dengan permasalahan statistik *multivariat* seperti *multipleregression*, analisa diskriminan, dan analisa kluster. Dalam banyak kasus, hasil yang didapat dengan ANN dapat dibandingkan dengan model statistika multivariat [8].



Gambar 2.5 Visualisasi proses ANN [8]

Dapat dilihat pada **Gambar 2.5** pada arsitektur jaringan ANN terdapat 3 buah layer yaitu *input layer*, *hidden layer*, dan *output layer*. *Input layer* sebagai masukan nilai yang hendak di training, dimana kemudian nilai masukan tersebut dikirim ke *hidden layer* dengan cara menghitung nya dengan bobot – bobot yang ada, kemudian dari *hidden layer* akan dikirim lagi ke *output layer* dengan cara yang sama yaitu dengan menghitung bobot – bobot yang ada. *Output layer* berisikan nilai dari target yang akan menjadi acuan pada tahap pengujian model (*testing*). Dari hasil pelatihan ini akan didapatkan model *neural network* yang sudah

terlatih dimana nilai bobot – bobot nya yang akan menentukan hasil dari prediksi ketika tahap pengujian [8].

2.6 Metode Evaluasi Kinerja

Proses klasifikasi yang dilakukan pada tugas akhir ini akan menghasilkan keluaran berupa kelas-kelas yang diidentifikasi pada setiap data yang diuji menggunakan ANN. Evaluasi terhadap performa atau kinerja ANN ini dilakukan dengan melakukan penghitungan nilai akurasi yang diperoleh.

Pada penempatan setiap data ke kelas yang diidentifikasi oleh sistem, akan dihitung jumlah proses identifikasi yang benar, yaitu ketika suatu data dengan kelas “1” diklasifikasikan sebagai kelas “1”, dan data kelas “2” diklasifikasi sebagai kelas “2”. Penghitungan akurasi didapatkan dengan menggunakan persamaan 2.3.

$$\text{Acc} = (T_0 + T_1) / N \quad 2.3$$

Pada persamaan 2.1 variabel hasil Acc merupakan hasil akurasi klasifikasi yang didapat, T_0 merupakan jumlah hasil klasifikasi benar untuk kelas “1”, T_1 merupakan hasil klasifikasi benar untuk kelas “2”, dan N merupakan jumlah data yang diuji atau diklasifikasikan.

Metode yang digunakan untuk melakukan evaluasi ini adalah *K-fold-cross validation*. Tujuan diimplementasikannya metode ini adalah untuk memperoleh hasil akurasi rata-rata, dari klasifikasi terhadap beberapa testing data berbeda yang terbagi dalam beberapa *fold*. Proses pengujian beberapa *fold* yang berbeda ini akan menghasilkan hasil yang diharapkan merepresentasikan performa sistem klasifikasi ini secara keseluruhan.

Pada *k-fold cross-validation*, pertama-tama keseluruhan dataset yang ada dibagi menjadi sebanyak k *fold*. Selanjutnya terhadap setiap *fold* dilakukan proses iterasi dimana *fold* ke- i akan berperan sebagai data *testing*, sedangkan $(k-1)$ *fold* sisanya menjadi data *training*. Iterasi pengujian dilakukan sebanyak k kali, sampai semua *fold* terpakai sebagai training untuk classifier. Pada setiap

iterasi dicatat hasil akurasi yang didapatkan, kemudian dicari rata-rata akhir akurasi dari kelima percobaan iterasi yang dilakukan.

Hasil akurasi akhir ini menunjukkan tingkat kesesuaian prediksi gerakan tangan yang dilakukan classifier ANN, berdasarkan input dataset EEG yang ada.

[Halaman ini sengaja dikosongkan]

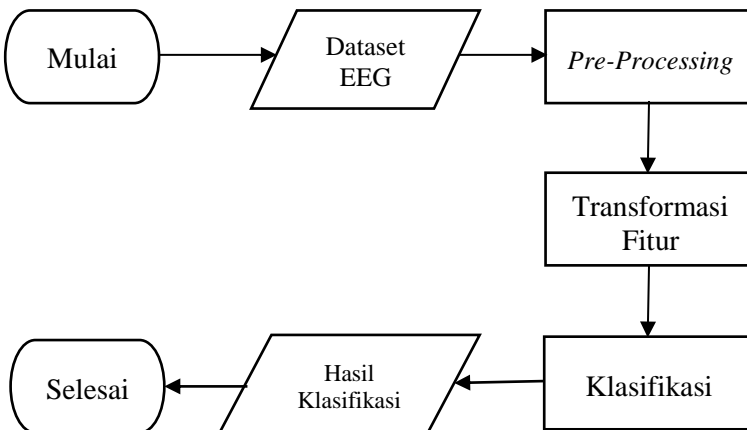
BAB III PERANCANGAN

Pada bab ini diuraikan mengenai perancangan aplikasi agar dapat mencapai tujuan dari tugas akhir. Perancangan yang dibuat meliputi perancangan sistem, data dan proses.

3.1 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan proses secara keseluruhan dari aplikasi deteksi pergerakan tangan dari data sinyal EEG. Untuk dapat menggunakan aplikasi deteksi pergerakan tangan ini, pengguna harus memiliki dataset sinyal EEG pergerakan tangan yang bisa didapatkan dari Dataset III BCI *Competition 2003* sebagai data masukan.

Sistem deteksi pada tugas akhir ini memiliki tiga proses utama. Proses pertama adalah *pre-processing*. Proses kedua adalah transformasi fitur. Proses ketiga adalah klasifikasi. Diagram alir dari sistem ditunjukkan pada **Gambar 3.1** berikut



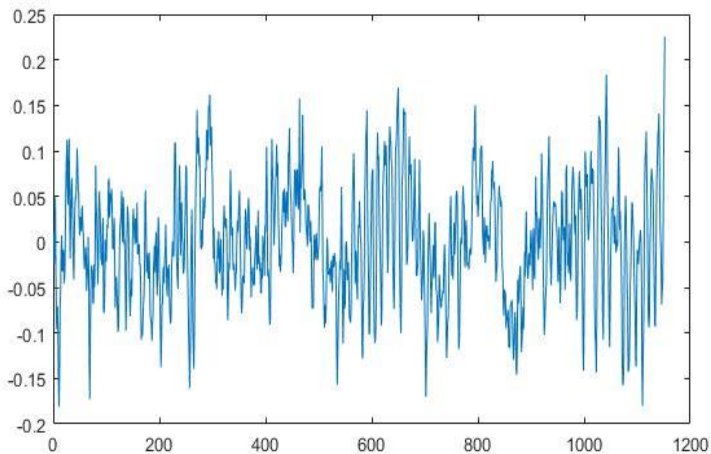
Gambar 3.1 Diagram Alir Sistem

Proses metode *pre-processing* adalah proses menyiapkan data signal yang akan digunakan pada proses transformasi fitur. Setelah

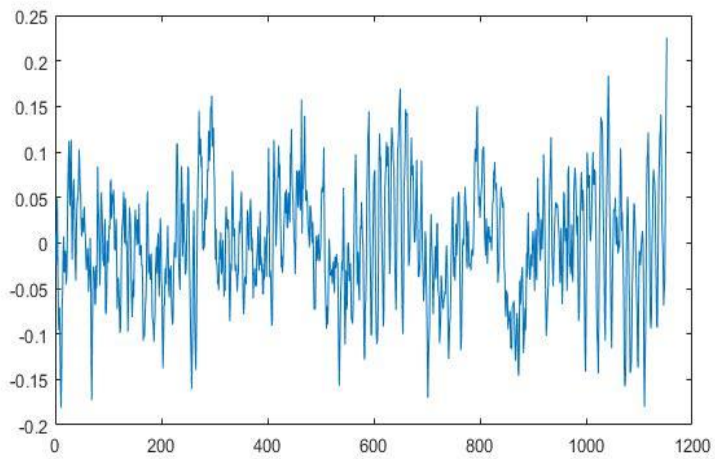
data masukan siap, dilanjutkan dengan proses ekstraksi fitur. Dalam tugas akhir ini penulis menggunakan metode transformasi fitur *Fast Walsh-Hadamart Transform* (FWHT). Dari data yang dihasilkan setelah proses transformasi fitur kemudian akan dilanjutkan dengan proses klasifikasi. Metode klasifikasi yang digunakan adalah *Artificial Neural Network* (ANN).

3.2 Perancangan Data

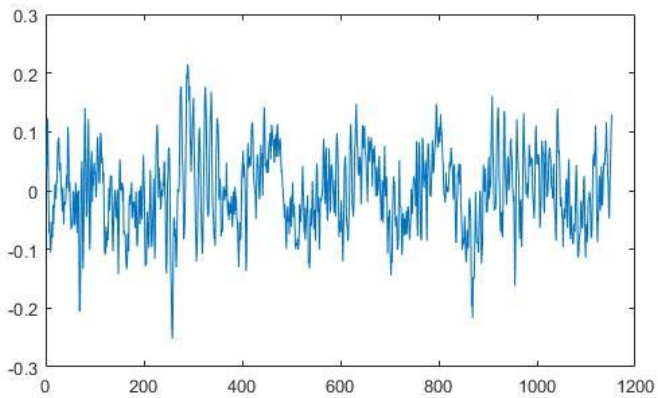
Data masukan pada sistem ini adalah sinyal EEG yang diperoleh dari *BCI Competition 2003 dataset III*. Sinyal yang tersedia, diambil dari hasil perekaman 3 *channel* elektroda yaitu: *Channel 1: C3, Channel 2: Cz, dan Channel 3: C4*. Untuk contoh data sinyal C3 ditunjukkan pada



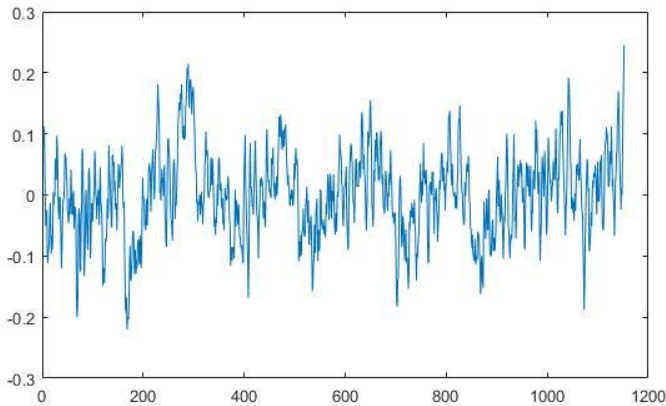
Gambar 3.2, contoh data sinyal Cz ditunjukkan pada **Gambar 3.3**, contoh data sinyal C4 ditunjukkan pada **Gambar 3.4**.



Gambar 3.2 Contoh Data Sinyal C3



Gambar 3.3 Contoh Data Sinyal Cz



Gambar 3.4 Contoh Data Sinyal C4

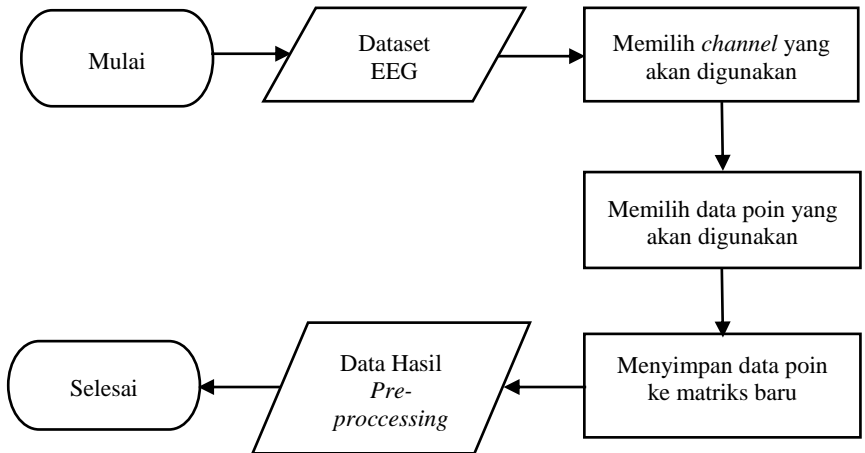
Matriks data mentah sinyal yang diperoleh berdimensi $N \times CT$, dimana N menandakan jumlah *trial*, C merupakan jumlah *channel*, dan T merupakan jumlah data poin untuk setiap *channel*. Terdapat total 280 *trial* pada dataset, dengan rincian 140 *trial* kelas 1 dan 140 *trial* kelas 2. Kelas 1 merepresentasikan gerakan tangan kiri, sedangkan kelas 2 merepresentasikan gerakan tangan kanan. Total data poin masing-masing *channel* berjumlah 1.152, yang berasal dari percobaan selama 9 detik dengan frekuensi 128 Hz.

3.3 Perancangan Proses

Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada aplikasi deteksi pergerakan tangan dari data sinyal EEG. Bagian dari setiap proses utama aplikasi dapat dilihat pada **Gambar 3.1**.

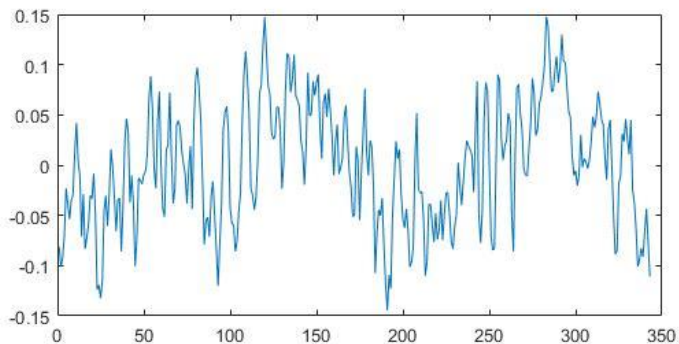
3.3.1 Pre-processing

Supaya aplikasi dapat berjalan optimal, diperlukan persiapan data terlebih dahulu pada proses *preprocessing*.

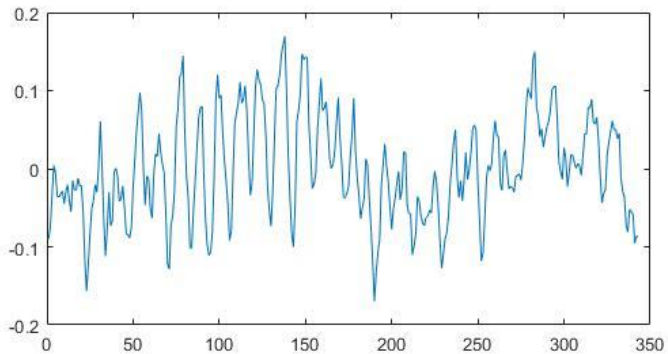


Gambar 3.5 Diagram Alir Preprocessing

Pada tugas akhir kali ini, tahap *preprocessing* yang dilakukan adalah memilih data poin signal EEG dan menyimpannya di sebuah array baru.

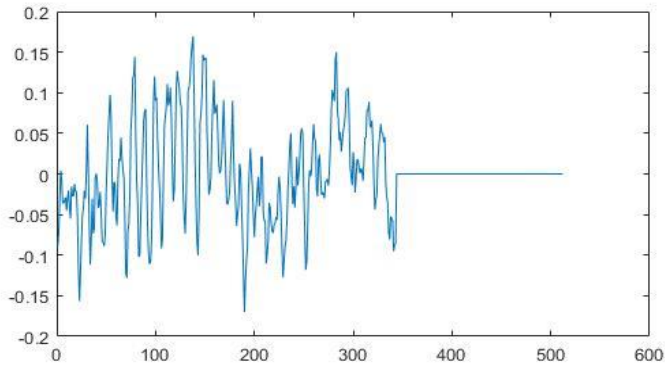


Gambar 3.6 Contoh Input Sinyal C3

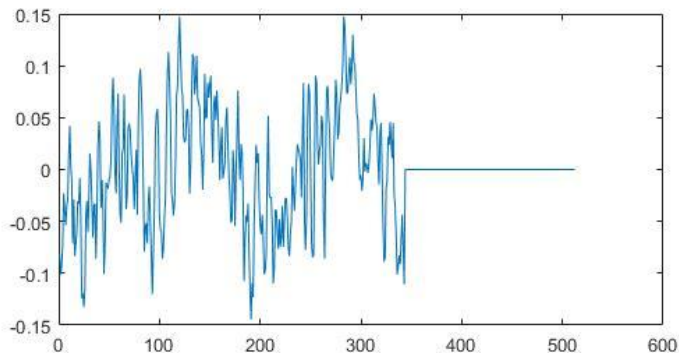


Gambar 3.7 Contoh Input Sinyal C4

Dari tiga channel yang ada di dataset EEG diambil hanya dua channel, yaitu channel C3 dan channel C4 [4]. Setiap channel terdiri dari 140 kali percobaan dimana setiap percobaan tadi masing-masing dilakukan selama 9 detik sehingga memiliki 1.152 data poin (128 data poin/detik). Dari 1.152 data poin tadi hanya data poin saat $t = 4s - 6,68s$ yang digunakan (sebanyak 343 data poin) [4]. Contoh input sinyal C3 seperti ditampilkan pada **Gambar 3.6** dan contoh input sinyal C4 seperti ditampilkan pada **Gambar 3.7**.



Gambar 3.8 Contoh Input Sinyal C3 Baru



Gambar 3.9 Contoh Input Sinyal C4 Baru

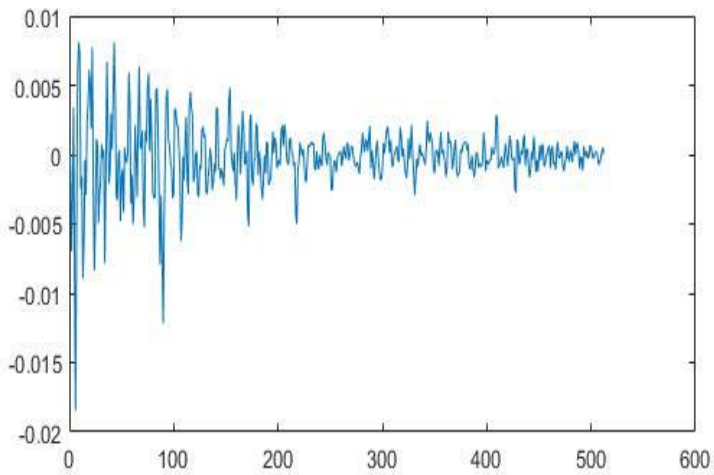
Untuk masuk ke tahap transformasi fitur, matriks data yang ada harus memiliki ukuran 2^n . Tetapi dikarenakan jumlah data dari data poin sinyal yang digunakan hanya 343 maka harus dibuat matriks baru yang menyimpan data poin tadi yang memiliki ukuran 2^n . Untuk itu dibuatlah matriks berukuran 512 dimana data poin tadi dimasukkan di awal dari matriks baru dan sisanya akan berisi nilai nol. Dari matriks baru ini lah data akan digunakan dalam tahap transformasi fitur. Contoh matriks baru untuk sinyal C3 seperti yang ditampilkan pada **Gambar 3.8** dan contoh matriks baru untuk

sinyal C4 seperti yang ditampilkan pada **Gambar 3.9**. sedangkan untuk tahapan untuk *preprocessing* dapat dilihat pada **Gambar 3.5**.

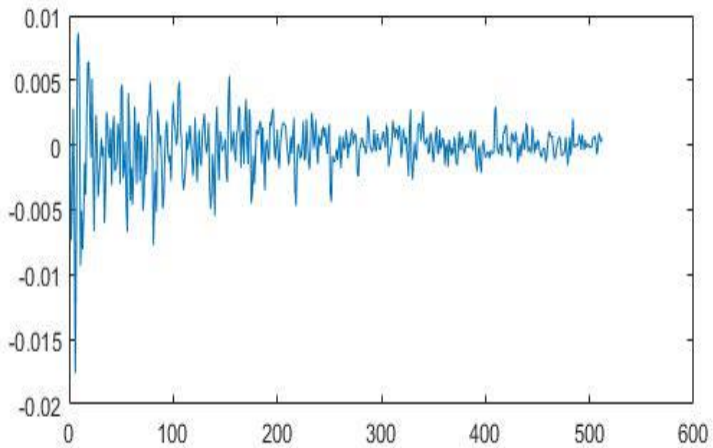
3.3.2 Transformasi Fitur

Tahap transformasi fitur pada tugas akhir ini dilakukan dengan metode *Fast Walsh Hadamart Transform* (FWHT). Hasil dari *output* tahap *preprocessing* akan digunakan sebagai *input* pada tahap ini. Dalam tahap ini dilakukan proses penambahan seperti yang ada pada **Gambar 2.4**. Data masukan pada tahap ini berukuran 512×280 *channel* C3 dan 512×280 *channel* C4. Dimana 512 adalah panjang sinyal masing-masing data dan 280 adalah banyaknya jumlah data yang diuji. Dari tahap transformasi fitur menggunakan FWHT data keluaran yang dihasilkan berukuran sama dengan data masukan yaitu 512×280 hasil transformasi fitur dari *channel* C3 dan 512×280 hasil transformasi fitur dari *channel* C4, dimana 512 adalah panjang sinyal hasil transformasi fitur dan 280 adalah banyaknya data yang diuji. Contoh hasil dari tahap transformasi fitur *channel* C3 seperti yang ditampilkan pada **Gambar 3.10** dan contoh hasil dari tahap transformasi fitur *channel* C4 seperti yang ditampilkan pada **Gambar 3.11**.

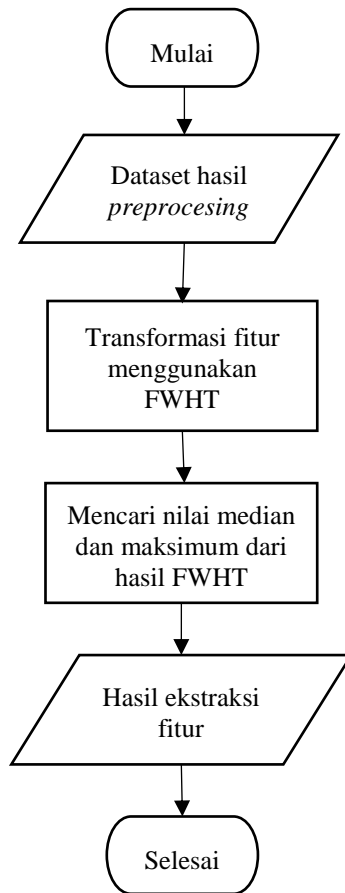
Setelah mendapatkan hasil FWHT dari masing-masing *channel*, dari hasil transformasi fitur tadi akan dicari nilai median dan juga nilai maksimum nya. Nilai median dan nilai maksimum inilah yang selanjutnya akan digunakan sebagai fitur dalam tahap klasifikasi. Fitur yang digunakan ini memiliki ukuran 4×280 dimana 4 merupakan banyaknya fitur (nilai median *channel* C3, mean *channel* C3, median *channel* C4, mean *channel* C4) dan 280 merupakan banyaknya data pada masing-masing fitur. Untuk proses ekstraksi fitur ini, tahapan lebih jelasnya ditampilkan oleh diagram alir **Gambar 3.12**.



Gambar 3.10 Hasil FWHT Channel C3



Gambar 3.11 Hasil FWHT Channel C4

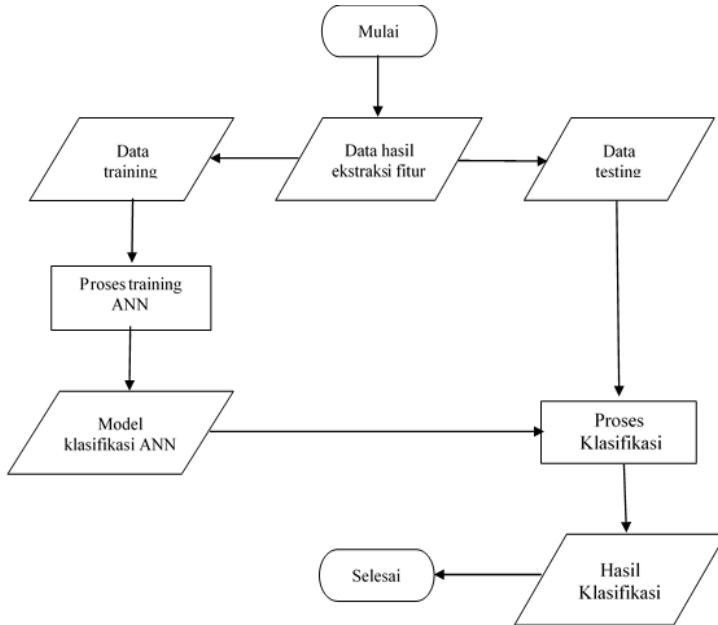


Gambar 3.12 Diagram Alir Transformasi fitur

3.3.3 Klasifikasi

Setelah melakukan proses transformasi fitur, selanjutnya akan dilakukan proses klasifikasi. Pada tugas akhir ini metode klasifikasi yang digunakan adalah *Artificial Neural Network* (ANN). Pada tahap ini dilakukan pengujian dengan nilai parameter *hidden layer*, jumlah neuron dan *learning rate* yang berbeda-beda. Hasil akhir dari tahap ini merupakan hasil kelas dari masing-

masing data yang diujikan. Tahapan pada proses ini dijelaskan di **Gambar 3.13**.



Gambar 3.13 Diagram Alir Klasifikasi

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1 Lingkungan Implementasi

Subbab ini akan menjelaskan mengenai lingkungan desain dan implementasi perangkat lunak yang akan dibangun. Spesifikasi perangkat keras yang digunakan dalam desain dan implementasi perangkat lunak adalah prosesor berjenis Intel(R) Core(TM) i5-7400M CPU @ 3.00 GHz, dengan kapasitas memori (RAM) sebesar 8.00 GB.

Sistem operasi yang digunakan adalah Microsoft Windows 10 Home Premium 64 Bit, sedangkan perangkat lunak yang digunakan dalam proses implementasi adalah MATLAB R2017a dan Spider.

4.2 Implementasi Program

Dalam subbab ini akan ditampilkan potongan kode program yang digunakan dalam klasifikasi sinyal EEG menggunakan *Fast Walsh–Hadamard Transform* dan *Artificial Neural Network* (ANN). Potongan kode program yang akan ditampilkan dalam subbab ini adalah potongan kode program untuk tahap *preprocessing*, transformasi fitur, dan klasifikasi

4.2.1 Implementasi Tahap *Preprocessing*

Pada tahap ini dilakukan pemilihan data signal EEG yang digunakan sebelum masuk ke proses transformasi fitur. Untuk tahap ini kode sumber ditampilkan pada **Kode sumber 4.1**

1. <code>for i=1:data</code>
2. <code> c3 = x_train(513:855,1,i);</code>
3. <code> temp1 = zeros(512,1);</code>
4. <code> temp1(1:length(c3)) = c3;</code>
5. <code> c4 = x_train(513:855,3,i);</code>
6. <code> temp2 = zeros(512,1);</code>
7. <code> temp2(1:length(c4)) = c4;</code>
8. <code>end</code>

Kode Sumber 4.1 Implementasi Tahap *Preprocessing*

Dari tiga buah channel yang ada, hanya channel C3 dan C4 yang digunakan. Dan dari 1152 data poin yang ada, hanya 343 data poin yang digunakan yaitu data poin ke 513 – 855 (disaat $t = 4s - 6.68s$). Baris 2 dan 5 kode sumber 4.1 adalah implementasi dari pengambilan data signal EEG C3 dan C4 dari dataset.

Untuk masuk ke proses transformasi fitur dibutuhkan 2^n data, maka dibuatlah variabel baru yang menyimpan data c3 dan c4 tadi dalam data 2^n yang terdekat, yaitu 512 data poin. Implementasi bisa dilihat dari kode sumber 4.1 baris 3-4 dan 6-7.

4.2.2 Implementasi Tahap Transformasi fitur

Setelah melakukan *preprocessing* data selanjutnya dilakukan tahap transformasi fitur. Disini transformasi fitur yang digunakan adalah metode *Fast Walsh Hadamart Transform* (FWHT). Untuk melakukan proses ini dilakukan pemanggilan fungsi *fhtseq* seperti **Kode Sumber 4.2.**

1. <code>for i=1:data</code>
2. <code> fwht1(:,i) = (fhtseq(temp1))/512;</code>
3. <code> fwht2(:,i) = (fhtseq(temp2))/512;</code>
4. <code>end</code>

Kode Sumber 4.2 Fungsi Pemanggilan FWHT

Untuk proses transformasi fitur di dalam fungsi *fhseq* sendiri terdapat pada **Kode Sumber 4.3**.

1. <code>x=bitrevorder(data);</code>
2. <code>N=length(x);</code>
3. <code>k1=N; k2=1; k3=N/2;</code>
4. <code>for i1=1:log2(N)</code>
5. <code>L1=1;</code>
6. <code>for i2=1:k2</code>
7. <code>for i3=1:k3</code>
8. <code>i=i3+L1-1; j=i+k3;</code>
9. <code>temp1= x(i); temp2 = x(j);</code>
10. <code>if(mod(i2,2) == 0)</code>
11. <code>x(i) = temp1 - temp2;</code>
12. <code>x(j) = temp1 + temp2;</code>
13. <code>else</code>
14. <code>x(i) = temp1 + temp2;</code>
15. <code>x(j) = temp1 - temp2;</code>
16. <code>end</code>
17. <code>end</code>
18. <code>L1=L1+k1;</code>
19. <code>end</code>
20. <code>k1 = k1/2; k2 = k2*2; k3 = k3/2;</code>
21. <code>end</code>

Kode Sumber 4.3 Implementasi Tahap Transformasi fitur

Dari **Kode Sumber 4.3** baris 1 berfungsi untuk mengembalikan input data matriks *x* dalam bentuk *bit-reversed* order. Baris 2-3 digunakan untuk batas awal dalam proses *addition* FWHT. Baris 4-21 adalah proses inti dari FWHT dimana terjadi proses *addition*.

Dari proses **Kode Sumber 4.2** akan menghasilkan matriks hasil FWHT dari channel C3 dan C4, *fwht1* adalah hasil

transformasi fitur channel C3 sedangkan fwht2 adalah hasil ekstraksi fitur channel C4.

1. %max dan median c3
2. for i=1:data
3. temp = (fwht1(:,i));
4. max1 (i) = max(temp);
5. median1 (i) = median(temp);
6. end
7. %max dan median c4
8. for i=1:data
9. temp = (fwht2(:,i));
10. max2 (:,i) = max(temp);
11. median2 (:,i) = median(temp);
12. end

Kode Sumber 4.4 Pengambilan fitur untuk klasifikasi

Setelah mendapatkan hasil transformasi fitur dari channel C3 dan C4 langkah selanjutnya adalah mencari nilai median dan juga nilai maksimumnya. Dari kode sumber 4.4 baris 2-6 berfungsi untuk mencari nilai median dan maksimum dari FWHT C3, baris 8-12 berfungsi untuk mencari nilai median dan maksimum dari FWHT C4. Nilai median dari FWHT C3 disimpan di matriks median1 dan nilai maksimum nya disimpan dalam matriks max1, Nilai median dari FWHT C4 disimpan di matriks median2 dan nilai maksimum nya disimpan dalam matriks max2.

4.2.3 Implementasi Klasifikasi

Sebelum masuk proses klasifikasi, fitur-fitur yang digunakan harus disimpan dalam satu matriks. Selain fitur-fitur yang digunakan, dalam matriks ini juga harus dimasukkan kelas dari masing-masing fitur tadi. Untuk penggabungan fitur ini langkah-langkahnya ada pada **kode sumber 4.5**.

1. features (:,1) = max1;
2. features (:,2) = median1;
3. features (:,3) = max2;
4. features (:,4) = median2;
5. features (:,5) = class;

Kode Sumber 4.5 Penggabungan Fitur untuk Tahap Klasifikasi

Dari **Kode Sumber 4.5** fitur-fitur yang digunakan untuk proses klasifikasi dan juga kelasnya disimpan dalam matriks baru yang bernama features. Baris 1 digunakan untuk memasukkan data max1 ke matriks features. Baris 2 digunakan untuk memasukkan data median1 ke matriks features. Baris 3 digunakan untuk memasukkan data max2 ke matriks features. Baris 4 digunakan untuk memasukkan data median2 ke matriks features. dan baris 5 digunakan untuk memasukkan data kelas ke matriks features.

1. k = 10;
2. C = crossvalind('Kfold',size(features,1),k);

Kode Sumber 4.6 Fungsi untuk Kfold Cross Validation

Sebelum masuk proses klasifikasi, metode kfold cross validation digunakan untuk evaluasi. Dari **Kode Sumber 4.6** baris 1 ditentukan nilai k=10, k adalah jumlah *fold*. Baris 2 adalah pemanggilan fungsi untuk membuat matriks *kfold cross validation*.

Setelah matriks C terbentuk, proses klasifikasi menggunakan *Artificial Neural Network (ANN)* bisa dilakukan. Proses klasifikasi ditunjukkan oleh **Kode Sumber 4.7** di bawah ini.

1. for z = 1:k
2. Test = (C==z);
3. Train = ~Test;
4.

5.	<code>traindata = features(Train,1:4).'</code>
6.	<code>trainclass = features(Train,5).'</code>
7.	<code>testdata = features(Test,1:4).'</code>
8.	<code>testclass = features(Test,5).'</code>
9.	<code>net = feedforwardnet([10]);</code>
10.	<code>net.trainparam.lr = 0,1;</code>
11.	<code>net = train(net,traindata,trainclass);</code>
12.	<code>hasil = round(sim(net,testdata));</code>
13.	
14.	<code>[m,n] = find(hasil==testclass);</code>
15.	<code>pre = sum(m)/size(testdata,2)*100;</code>
16.	<code>akurasi (z) = pre;</code>
17.	<code>end</code>
18.	<code>mean (akurasi)</code>

Kode Sumber 4.7 Implementasi Untuk Tahap Klasifikasi Menggunakan ANN

1.	<code>#Random Forest#</code>
2.	<code>clf =</code>
3.	<code>RandomForestClassifier(n_estimators=100,</code>
4.	<code>oob_score=True,</code>
5.	<code>random_state=np.random.RandomState())</code>
6.	<code>#SVM#</code>
7.	<code>clf = svm.SVC()</code>
8.	<code>#Naive Bayes#</code>
9.	<code>clf = GaussianNB()</code>

Kode Sumber 4.8 Implementasi Klasifikasi Menggunakan Metode Naive Bayes, SVM, dan Random Forest

Dari **Kode Sumber 4.7** baris 2 dan 3 berfungsi untuk membagi data *testing* dan data *training* dari matriks features berdasarkan indeks yang terdapat di matriks C. Setelah data untuk *train* dan *test* sudah terbagi, selanjutnya ditentukan kolom yang

merupakan fitur klasifikasi dan kelas, untuk melakukan pembagian ini dilakukan dengan cara seperti pada baris 5-6. Baris 9 berfungsi untuk membuat network baru, dimana *hidden layer* berjumlah 1 dengan jumlah neuron 10 dan *learning function* 'trainlm'. Baris 10 digunakan untuk menentukan besaran *learning rate* yang digunakan.

Setelah network terbentuk dilakukan proses *train network* untuk klasifikasi seperti pada baris 11. Baris 12 dan 13 digunakan untuk menentukan kelas hasil *train network* tadi dan mencocokkan dengan kelas sebenarnya. Pada baris 15, Pre adalah hasil akurasi dari proses klasifikasi *fold* ke-z. dan pada baris 16, akurasi adalah sebuah matriks yang menyimpan hasil akurasi klasifikasi setiap *fold* ke-z tadi. Rata-rata akurasi dari 10 *fold* bisa ditampilkan dengan menggunakan kode pada baris ke 18.

Selain klasifikasi menggunakan metode klasifikasi ANN, dilakukan juga klasifikasi dengan metode lain sebagai pembandingan. Pada **Kode Sumber 4.8** baris 2-5 adalah pemanggilan fungsi untuk klasifikasi menggunakan metode Random Forest, baris 7 untuk klasifikasi menggunakan metode SVM, dan baris 9 pemanggilan fungsi untuk metode klasifikasi.

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

Dalam proses pengujian sistem, dibutuhkan suatu lingkungan uji coba yang sesuai dengan standar kebutuhan. Uji coba aplikasi ini dilakukan dengan menggunakan *personal computer* (PC). Berikut ini spesifikasi PC yang digunakan.

- Perangkat keras
 - Intel® Core™ i5-7400S CPU @ 3.00 GHz
 - *Installed Memory* (RAM) 8,00 GB
- Perangkat lunak
 - Windows 10
 - MATLAB R2017a
 - Microsoft Office.
 - Spider

5.2 Data Uji Coba

Data yang digunakan untuk uji coba klasifikasi sinyal EEG dengan menggunakan metode transformasi fitur *Fast Walsh Hadamart Transform* serta *classifier* yang menggunakan metode *Artificial Neural Network* (ANN) ini diperoleh dari BCI *Competition* 2003 dataset III. Data awal yang didapat berupa matriks sinyal dengan dimensi $N \times CT$. N merupakan jumlah *trial* yaitu sebanyak 280, dengan rincian 140 *trial* pertama merupakan kelas 1 sedangkan 140 *trial* selanjutnya merupakan kelas 2. Kelas 1 pada dataset ini merepresentasikan gerakan tangan kanan, sedangkan kelas 2 merepresentasikan gerakan tangan kiri. C merepresentasikan jumlah *channel*, T merupakan jumlah data poin untuk setiap *channel* yang direkam pada dataset 1a ini. Untuk

setiap *channel*, terdapat 1.152 data poin, dan jumlah *channel* secara keseluruhan adalah sebanyak 3 *channel*.

Metode yang digunakan untuk skenario uji coba ini adalah *k-fold cross validation*, dengan *k* yang dipilih sejumlah 10 buah. Untuk kepentingan ini, *trial* di masing-masing kelas akan dibagi ke dalam 10 *fold*.

5.3 Skenario Uji Coba

Uji coba dilakukan untuk mengetahui nilai-nilai parameter yang tepat untuk digunakan dalam proses deteksi pergerakan tangan dari data sinyal EEG. Nilai parameter yang tepat penting untuk diketahui karena penggunaan parameter yang tepat akan memberikan hasil yang terbaik dalam proses klasifikasi dan akan memberikan hasil akurasi yang tinggi.

Dalam tugas akhir ini, skenario pengujian terdiri dari empat macam yaitu:

1. Uji coba penentuan parameter pada jumlah neuron dengan satu *hidden layer* terhadap akurasi.
2. Uji coba penentuan parameter pada jumlah neuron dengan dua *hidden layer* terhadap akurasi.
3. Uji coba penentuan parameter *learning rate* dengan satu *hidden layer* terhadap akurasi.
4. Uji coba penentuan parameter *learning rate* dengan dua *hidden layer* terhadap akurasi.
5. Uji coba perbandingan metode klasifikasi ANN dengan metode klasifikasi *Random Forest*, *Naïve Bayes* dan SVM.

5.3.1 Skenario Uji Coba 1

Skenario uji coba 1 adalah pengaruh penentuan parameter pada jumlah neuron dengan satu *hidden layer* terhadap akurasi. Skenario dicoba dengan menggunakan beberapa variasi parameter jumlah neuron, yaitu 5, 10, 20, 35, dan 50, dimana akan dilakukan *running* program sebanyak lima kali pada masing-masing parameter. Untuk itu setiap parameter akan memiliki lima akurasi

dan juga satu akurasi rata-rata yang dihasilkan. Lima kali percobaan ini dilakukan karena walaupun memiliki nilai parameter yang sama akurasi yang dihasilkan setiap kali program dijalankan akan selalu berbeda-beda. Untuk mendapatkan hasil yang lebih akurat maka dari itu program dijalankan sebanyak lima kali dan diambil rata-ratanya. Dari hasil rata-rata akurasi inilah akan dicari hasil akurasi tertinggi.

Setiap satu kali percobaan dilakukan dengan metode *k-fold cross validation* dengan nilai $k = 10$. Persentase akurasi tiap percobaan yang ditampilkan adalah rata-rata hasil akurasi yang didapatkan melalui metode *k-fold cross validation*. Hasil dari skenario uji coba 1 dapat dilihat pada Tabel 5.1 berikut.

Tabel 5.1 Hasil Akurasi Skenario Uji Coba 1

Jumlah Neuron	Akurasi (%) Percobaan ke-					Rata-Rata Akurasi (%)
	1	2	3	4	5	
5	72,5	68,22	73,21	70,72	68,93	70,72
10	72,14	72,50	71,43	73,57	71,43	72,21
20	69,29	70,36	68,57	70,72	71,79	70,14
35	67,50	72,14	70,72	72,50	73,22	71,22
50	66,79	64,29	67,86	66,07	69,27	66,86

Dari Tabel 5.1 dapat diketahui bahwa aplikasi yang memiliki jumlah jumlah neuron 10 menghasilkan nilai rata-rata akurasi yang paling tinggi yaitu 72,21%. Untuk hasil akurasi tiap *fold* dari akurasi yang memiliki jumlah neuron 10 serta hasil prediksi tiap data tes dari akurasi tertinggi akan ditampilkan pada lampiran.

5.3.2 Skenario Uji Coba 2

Skenario uji coba 2 adalah pengaruh penentuan parameter pada jumlah neuron dengan dua *hidden layer* terhadap akurasi. Skenario dicoba dengan menggunakan beberapa variasi parameter

jumlah neuron, yaitu [5 5], [10 10], [20 20], [35 35], dan [50 50], dimana akan dilakukan *running* program sebanyak lima kali pada masing-masing parameter. Hasilnya setiap parameter akan memiliki lima akurasi dan juga satu akurasi rata-rata. Seperti pada skenario uji coba 1, lima kali percobaan ini dilakukan karena walaupun memiliki nilai parameter yang sama akurasi yang dihasilkan setiap kali program dijalankan akan selalu berbeda-beda. Untuk mendapatkan hasil yang lebih akurat maka dari itu program dijalankan sebanyak lima kali dan diambil rata-ratanya. Dari hasil rata-rata akurasi inilah akan dicari hasil akurasi tertinggi.

Setiap satu kali percobaan dilakukan dengan metode *k-fold cross validation* dengan nilai $k = 10$. Persentase akurasi tiap percobaan yang ditampilkan adalah rata-rata hasil akurasi yang didapatkan melalui metode *k-fold cross validation*. Hasil dari skenario uji coba 2 dapat dilihat pada Tabel 5.2 berikut.

Tabel 5.2 Hasil Akurasi Skenario Uji Coba 2

Jumlah Neuron	Akurasi (%) Percobaan ke-					Rata-Rata Akurasi (%)
	1	2	3	4	5	
[5 5]	76,07	76,43	70,36	73,93	70,00	73,36
[10 10]	70,00	71,43	73,99	71,43	70,36	71,43
[20 20]	68,93	69,29	69,64	68,21	69,29	69,07
[35 35]	64,29	65,36	68,93	66,43	70,00	67,00
[50 50]	63,93	64,64	58,21	64,29	62,86	62,78

Dari Tabel 5.2 dapat diketahui bahwa aplikasi yang memiliki jumlah jumlah neuron [5 5] menghasilkan nilai rata-rata akurasi yang paling tinggi yaitu 73,36%. Untuk hasil akurasi tiap *fold* dari akurasi yang memiliki jumlah neuron [5 5] serta hasil prediksi tiap data tes dari akurasi tertinggi akan ditampilkan pada lampiran.

5.3.3 Skenario Uji Coba 3

Skenario uji coba 3 adalah pengaruh penentuan parameter pada *learning rate* dengan satu *hidden layer* dengan jumlah neuron 10 terhadap akurasi. Skenario dicoba dengan menggunakan beberapa variasi parameter *learning rate*, yaitu 0,001; 0,005; 0,01; 0,05; dan 0,1 dimana akan dilakukan *running* program sebanyak lima kali pada masing-masing parameter. Hasilnya setiap parameter akan memiliki lima akurasi dan juga satu akurasi rata-rata. Lima kali percobaan ini dilakukan karena walaupun memiliki nilai parameter yang sama akurasi yang dihasilkan setiap kali program dijalankan akan selalu berbeda-beda. Untuk mendapatkan hasil yang lebih akurat maka dari itu program dijalankan sebanyak lima kali dan diambil rata-ratanya. Dari hasil rata-rata akurasi inilah akan dicari hasil akurasi tertinggi.

Setiap satu kali percobaan dilakukan dengan metode *k-fold cross validation* dengan nilai $k = 10$. Persentase akurasi tiap percobaan yang ditampilkan adalah rata-rata hasil akurasi yang didapatkan melalui metode *k-fold cross validation*. Hasil dari skenario uji coba 2 dapat dilihat pada Tabel 5.3 berikut.

Tabel 5.3 Hasil Akurasi Skenario Uji Coba 3

<i>Learning Rate</i>	Akurasi (%) Percobaan ke-					Rata-Rata Akurasi (%)
	1	2	3	4	5	
0,001	69,64	70,72	69,29	75,00	72,86	71,50
0,005	71,43	70,72	73,21	71,43	70,72	71,49
0,01	62,10	71,07	73,57	73,22	72,50	71,57
0,05	71,79	73,21	71,79	72,14	73,57	72,50
0,1	72,14	72,50	71,43	73,57	71,43	72,21

Dari Tabel 5.3 didapatkan hasil rata-rata akurasi tertinggi saat nilai *learning rate* nya 0,05 dimana nilai akurasinya sebesar

72,50%. Untuk hasil akurasi tiap *fold* yang memiliki nilai *learning rate* 0,05 serta hasil prediksi tiap data tes dari akurasi tertinggi akan ditampilkan pada lampiran.

5.3.4 Skenario Uji Coba 4

Skenario uji coba 4 adalah pengaruh penentuan parameter pada *learning rate* dengan dua *hidden layer* dengan jumlah jumlah neuron [5 5] terhadap akurasi. Skenario dicoba dengan menggunakan beberapa variasi parameter *learning rate*, yaitu 0,001; 0,005; 0,01; 0,05; dan 0,1 dimana akan dilakukan *running* program sebanyak lima kali pada masing-masing parameter. Untuk itu setiap parameter akan memiliki lima akurasi dan juga satu akurasi rata-rata yang dihasilkan. Lima kali percobaan ini dilakukan karena walaupun memiliki nilai parameter yang sama akurasi yang dihasilkan setiap kali program dijalankan akan selalu berbeda-beda. Untuk mendapatkan hasil yang lebih akurat maka dari itu program dijalankan sebanyak lima kali dan diambil rata-ratanya. Dari hasil rata-rata akurasi inilah akan dicari hasil akurasi tertinggi.

Tabel 5.4 Hasil Akurasi Skenario Uji Coba 4

<i>Learning Rate</i>	Akurasi Percobaan ke-					Rata-Rata Akurasi
	1	2	3	4	5	
0,001	70,00	67,86	73,22	69,64	71,79	70,50
0,005	73,93	70,358	72,50	73,57	71,07	72,29
0,01	71,78	67,1	70,71	71,07	72,50	70,63
0,05	70,36	70,00	73,22	70,36	72,50	71,29
0,1	75,00	72,86	76,07	70,72	74,29	73,79

Setiap satu kali percobaan dilakukan dengan metode *k-fold cross validation* dengan nilai $k = 10$. Persentase akurasi tiap percobaan yang ditampilkan adalah rata-rata hasil akurasi yang

didapatkan melalui metode *k-fold cross validation*. Hasil dari skenario uji coba 2 dapat dilihat pada Tabel 5.4.

Dari Tabel 5.4 didapatkan hasil rata-rata akurasi tertinggi saat nilai *learning rate* nya 0,1 dimana nilai akurasinya sebesar 73,79%. Untuk hasil akurasi tiap *fold* yang memiliki nilai *learning rate* 0,1 serta hasil prediksi tiap data tes dari akurasi tertinggi akan ditampilkan pada lampiran.

5.3.5 Skenario Uji Coba 5

Pada skenario uji coba 5 dilakukakn perbandingan hasil akurasi menggunakan tiga metode klasifikasi lain, yaitu Random Forest, Naïve Bayes, dan Support Vector Machines (SVM). Pada masing-masing metode dilakukan lima kali *running* program, dimana pada masing-masing iterasi dilakukan metode *k-fold cross validation* dengan nilai $k = 10$.

Tabel 5.5 Hasil Akurasi Skenario Uji Coba 5

Metode Klasifikasi	Rata-Rata Akurasi (%)
Random Forest	65,93
SVM	73,71
Naïve Bayes	71,00
ANN	73,79

Dari **Tabel 5.5** proses klasifikasi menggunakan metode ANN memiliki nilai rata-rata akurasi tertinggi yaitu 73,79%. Untuk itu, metode klasifikasi ANN merupakan metode yang lebih optimal jika dibandingkan dengan metode Random Forest, Naïve Bayes, dan Support Vector Machines (SVM). Untuk hasil akurasi untuk tiap iterasi akan ditampilkan pada lampiran.

5.3.6 Skenario Uji Coba 6

Berdasarkan *paper* acuan yang digunakan dalam tugas akhir ini, fitur yang digunakan ada 4, yaitu median C3, max C3, median C4, dan max C4. Untuk itulah pada skenario uji coba ini dilakukan perbandingan hasil akurasi jika fitur yang digunakan 6 fitur, yaitu fitur awal ditambah dengan nilai mean dari C3 dan C4.

Tabel 5.6 Hasil Akurasi Skenario Uji Coba 6

Jumlah Fitur	Akurasi Percobaan ke-					Rata-Rata Akurasi
	1	2	3	4	5	
4 Fitur	75,00	72,86	76,07	70,72	74,29	73,79
6 Fitur	56,8	66,43	64,64	63,21	60,71	62,36

Dari hasil uji coba yang dilakukan pada Tabel 5.6, proses klasifikasi menggunakan 4 fitur menghasilkan nilai rata-rata akurasi tertinggi yaitu 73,79%.

5.3.7 Skenario Uji Coba 7

Berdasarkan *paper* acuan yang digunakan dalam tugas akhir ini, untuk mendapatkan fitur dilakukan pemotongan sinyal sepanjang 343 data poin. Untuk itulah pada skenario uji coba ini dilakukan perbandingan hasil akurasi jika sinyal dipotong sepanjang 512 data poin.

Dari hasil uji coba pada Tabel 5.7 **Kesalahan! Sumber referensi tidak ditemukan.**, hasil klasifikasi ketika sinyal dipotong sepanjang 343 data poin menghasilkan nilai rata-rata akurasi lebih tinggi jika dibandingkan dengan ketika sinyal dipotong sepanjang 512 data poin. Nilai rata-rata akurasi tertinggi yang bias didapatkan yaitu 73,79%.

Tabel 5.7 Hasil Akurasi Skenario Uji Coba 7

Jumlah Data Poin	Akurasi Percobaan ke-					Rata-Rata Akurasi
	1	2	3	4	5	
343	75,00	72,86	76,07	70,72	74,29	73,79

512	64,29	62,86	62,14	59,64	60,71	61,93
-----	-------	-------	-------	-------	-------	-------

5.4 Evaluasi Hasil Uji Coba

Berdasarkan hasil uji coba yang telah dilakukan, penentuan jumlah *hidden layer* dan jumlah neuron pada proses klasifikasi ANN memberikan nilai rata-rata akurasi yang lebih baik dengan dua *hidden layer* dan dengan jumlah jumlah neuron [5 5], dimana rata-rata akurasi yang dihasilkan adalah 73,36%. Ketika proses klasifikasi hanya menggunakan satu *hidden layer*, rata-rata akurasi tertinggi yang dihasilkan adalah 72,21%, dimana didapatkan pada saat jumlah neuron-nya bernilai 10.

Uji coba selanjutnya yang dilakukan adalah merubah parameter pada *hidden layer* dan jumlah neuron, dilakukan juga uji coba dengan merubah nilai parameter pada *learning rate*-nya. Setelah dilakukan uji coba menggunakan satu *hidden layer* didapatkan akurasi tertinggi dengan jumlah neuron bernilai 10 dan dengan menggunakan dua *hidden layer* didapatkan akurasi tertinggi dengan jumlah neuron bernilai [5 5]. Sehingga pada uji coba ini untuk satu *hidden layer* digunakan jumlah neuron 10 sebagai *default* dan untuk dua *hidden layer* digunakan jumlah neuron [5 5] sebagai *default*.

Menggunakan satu buah *hidden layer* didapatkan akurasi tertinggi sebesar 72,50% ketika parameter pada *learning rate* bernilai 0,05. Sedangkan saat menggunakan dua buah *hidden layer* didapatkan akurasi tertinggi sebesar 73,79% ketika parameter pada *learning rate* bernilai 0,1. Akurasi 73,79% merupakan akurasi tertinggi yang didapatkan, untuk itulah hasil ini akan digunakan untuk uji coba selanjutnya.

Pada uji coba 5 dilakukan perbandingan nilai akurasi yang didapatkan metode klasifikasi ANN dengan metode klasifikasi lain, yaitu metode Random Forest, Naïve Bayes, dan Support Vector Machines (SVM). Berdasarkan hasil yang didapatkan, aplikasi yang menggunakan metode klasifikasi ANN mendapatkan rata-rata akurasi tertinggi yaitu 73,79%.

Pada uji coba 5 dan 6 dilakukan perbandingan nilai akurasi jika digunakan 6 fitur serta pemotongan sinyal sepanjang 512 data poin. Dari hasil percobaan penggunaan 4 fitur (median C3, max C3, median C4, dan max C4) menghasilkan nilai akurasi yang lebih tinggi daripada ketika menggunakan 6 fitur (median C3, max C3, mean C3, median C4, max C4, dan mean C4). Begitu pula saat dilakukan pemotongan sinyal sepanjang 512 data poin, hasil akurasi yang didapatkan tidak lebih baik jika dilakukan pemotongan sinyal sepanjang 343 data poin.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Dari hasil pengamatan yang diperoleh dari uji coba dan evaluasi yang dilakukan, maka dapat disimpulkan sebagai berikut.

1. Parameter jumlah jumlah neuron terbaik dengan satu *hidden layer* adalah 20, dimana didapatkan hasil rata-rata akurasi tertinggi 72,21%.
2. Parameter jumlah jumlah neuron terbaik dengan dua *hidden layer* adalah [5 5], dimana didapatkan hasil rata-rata akurasi tertinggi 73,36%.
3. Parameter nilai *learning rate* terbaik untuk satu *hidden layer* dengan jumlah neuron 20 adalah 0,05, dimana didapatkan hasil rata-rata akurasi tertinggi 72,50%.
4. Parameter nilai *learning rate* terbaik untuk dua *hidden layer* dengan jumlah neuron [5 5] adalah 0,1, dimana didapatkan hasil rata-rata akurasi tertinggi 73,79%.
5. Metode klasifikasi ANN memberikan hasil akurasi yang lebih baik dibandingkan dengan metode *random forest*, *naïve bayes* dan SVM.
6. Klasifikasi menggunakan 4 fitur menghasilkan rata-rata nilai akurasi lebih tinggi daripada menggunakan 6 fitur.
7. Nilai akurasi yang didapatkan dengan pemotongan sinyal sepanjang 343 data poin lebih tinggi jika dibandingkan dengan pemotongan sinyal sepanjang 512 data poin.

6.2 Saran

Saran-saran yang dapat diberikan dalam pengujian pergerakan tangan kanan/kiri dari data sinyal EEG adalah:

1. Menggunakan metode transformasi fitur lain, sehingga didapatkan fitur-fitur yang bisa mewakili masing-masing kelas sehingga bisa memudahkan dalam proses klasifikasi dan bisa menghasilkan akurasi yang lebih baik.
2. Melakukan eksplorasi parameter pada metode klasifikasi ANN.
3. Menggunakan metode klasifikasi lain yang dapat menaikkan akurasi.

DAFTAR PUSTAKA

- [1] F. Yonghui, M. Chen, and X. Zheng, "Extracting features from phase space of EEG signals in brain-computer interfaces." *Neurocomputing* 151, 1477-1485, 2015.
- [2] B. Alessio Paolo, H.O. Keles, and A. Omurtag, "Hybrid EEG-fNIRS Asynchronous Brain-Computer Interface for Multiple Motor Tasks." *PloS one* 11.1, e0146610, 2016.
- [3] M. Zhanyu, Z.H. Tan and J. Guo, "Feature selection for neutral vector in EEG signal Classification." *Neurocomputing* 174, 937-945, 2016.
- [4] Kübra Saka, Önder Aydemir, Mehmet Öztürk "Classification of EEG Signals Recorded During Right/Left Hand Movement Imagery Using Fast Walsh Hadamard Transform Based Features." Scientific and Technological Research Council of Turkey (TUBITAK), Project No: EEEAG-215E155, 2016
- [5] Y., Weibo, et al. "EEG oscillatory patterns and classification of sequential compound limb motor imagery." *Journal of neuroengineering and rehabilitation*, 13.1, 1, 2016.
- [6] E. Niedermeyer and L. da Silva F. "Electroencephalography: Basic Principles, Clinical Applications, and Related Fields." Lippincot Williams & Wilkins. ISBN 0-7817-5126-8, 2004.
- [7] " Elektro Enselo Grafi (EEG)" 28 Frebruary 2011. [Online]. Available: <http://akatsuki-ners.blogspot.co.id/2011/02/elektro-enselografi-eeg.html/>. [Accessed 5 January 2018].
- [8] C. Stergiou dan D. Sigano, "Neural Network," [Online]. Available: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html. [Accessed 30 December 2017].
- [9] "Fast Walsh-Hadamard Transform (Sequency Ordered)" [Online]. Available: <http://fourier.eng.hmc.edu/e161/lectures/wht/node4.html>. [Accessed 2 Februari 2018].
- [10] "Fast Walsh-Hadamard Transform" [Online]. Available: https://en.wikipedia.org/wiki/Fast_Walsh%E2%80%9393Hadamard_transform. [Accessed 2 Februari 2018].

- [11 T. L. Rusch, R. Sankar, J. E. Scharf, , “Signal processing
] methods for pulse oximetry”, *Computer in Biology and Medicine*,
vol. 26, Issue 2, pp 143-159, 1999
- [12 A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood
] Cliffs, NJ, USA: Prentice-Hall, 1989
- [13 Fino, B. J.; Algazi, V. R. (1976). "Unified Matrix Treatment of the
] Fast Walsh–Hadamard Transform". *IEEE Transactions on
Computers*. 25 (11): 1142–1146
- [14 Fitriyani , Putri Nur “Deteksi Penyakit Epilepsi Berdasarkan Data
] EEG Otak Manusia Menggunakan Empirical Mode Decomposition
dan Interval Type-2 Fuzzy Logic System”. 2017.
- [15 “Hadamard Transform (Sequency Ordered)” [Online]. Available:
] https://en.wikipedia.org/wiki/Hadamard_transform. [Accessed 2
Juli 2018].

LAMPIRAN

- Hasil dari Skenario Uji Coba 1
 1. Tabel hasil akurasi tiap *fold* yang didapatkan dengan satu *hidden layer* dan *jumlah neuron* 10.

Fold ke-	Akurasi (%) Uji Coba ke-				
	1	2	3	4	5
1	75,00	71,43	71,43	78,57	64,29
2	67,86	64,29	60,71	71,43	64,29
3	60,71	67,86	60,71	64,29	53,57
4	82,14	71,43	75,00	60,71	78,57
5	71,43	75,00	78,57	78,57	78,57
6	78,57	78,57	89,29	85,71	85,71
7	71,43	71,43	75,00	78,57	64,29
8	67,86	75,00	60,71	78,57	82,14
9	78,57	82,14	75,00	75,00	78,57
10	67,86	67,86	67,86	64,29	64,29
Rata-rata	72,14	72,50	71,43	73,57	71,43

2. Tabel hasil prediksi tiap *fold* dengan *truth class* dari rata-rata akurasi tertinggi uji coba dengan satu *hidden layer* dan jumlah neuron 10.

Fold 1			Fold2		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	1	2	1	2	1
2	2	2	2	1	1
3	2	2	3	1	1
4	2	2	4	1	2
5	2	2	5	1	2
6	1	1	6	2	2
7	1	1	7	1	2
8	1	2	8	2	2
9	2	1	9	2	2
10	2	2	10	1	1
11	1	1	11	1	1
12	1	2	12	2	1
13	1	1	13	2	1
14	2	1	14	2	2
15	2	2	15	1	2
16	2	2	16	2	2
17	2	2	17	2	2
18	2	2	18	1	1
19	1	1	19	2	2
20	1	1	20	2	2
21	2	2	21	1	2
22	1	1	22	2	2
23	2	2	23	2	2
24	2	1	24	1	1
25	1	1	25	1	1
26	2	2	26	1	1
27	2	2	27	2	2
28	2	2	28	2	2

Fold3			Fold4		
Data Tes ke-	Aktual	Prediksi	Data Tes ke	Aktual	Prediksi
1	1	1	1	2	1
2	2	1	2	2	2
3	1	1	3	2	1
4	1	2	4	2	1
5	2	1	5	1	1
6	1	1	6	1	2
7	2	2	7	1	1
8	2	2	8	2	2
9	1	2	9	1	1
10	1	1	10	2	1
11	1	1	11	2	1
12	2	2	12	1	1
13	2	2	13	2	1
14	2	1	14	1	1
15	1	1	15	2	2
16	1	2	16	2	2
17	2	1	17	2	2
18	1	1	18	2	2
19	1	1	19	2	2
20	1	1	20	2	2
21	2	1	21	1	1
22	2	2	22	2	1
23	2	2	23	1	2
24	1	1	24	1	1
25	1	2	25	2	1
26	2	1	26	2	1
27	2	2	27	1	1
28	1	1	28	1	1

Fold5			Fold6		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	1	2	1	2	1
2	2	1	2	1	1
3	1	1	3	2	2
4	2	2	4	1	1
5	1	1	5	2	2
6	1	2	6	2	2
7	1	1	7	1	1
8	2	2	8	1	1
9	2	2	9	1	1
10	1	1	10	1	1
11	1	1	11	2	2
12	2	2	12	1	1
13	1	1	13	1	1
14	2	2	14	1	1
15	1	1	15	2	2
16	2	2	16	1	1
17	2	2	17	1	1
18	2	2	18	1	1
19	1	1	19	1	1
20	2	2	20	1	2
21	1	1	21	1	1
22	2	1	22	2	2
23	2	2	23	2	2
24	2	2	24	1	2
25	2	1	25	2	1
26	1	1	26	2	2
27	1	1	27	1	1
28	2	1	28	2	2

Fold8			Fold7		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	2	2	1	2	2
2	2	2	2	2	2
3	1	1	3	1	1
4	2	2	4	1	2
5	2	2	5	1	1
6	2	2	6	2	2
7	2	2	7	1	1
8	2	2	8	1	1
9	1	2	9	2	2
10	1	1	10	2	2
11	1	2	11	2	2
12	2	1	12	1	2
13	1	1	13	2	2
14	2	2	14	2	2
15	2	2	15	1	1
16	1	1	16	1	2
17	1	1	17	2	2
18	1	2	18	1	1
19	2	2	19	1	2
20	1	1	20	1	1
21	1	1	21	1	1
22	1	1	22	1	2
23	2	2	23	2	1
24	2	2	24	1	1
25	2	1	25	2	2
26	1	1	26	1	1
27	2	1	27	1	1
28	1	1	28	2	2

Fold9			Fold10		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	2	2	1	2	2
2	1	2	2	2	2
3	1	1	3	2	2
4	2	1	4	1	1
5	2	2	5	2	1
6	1	2	6	1	1
7	2	2	7	2	2
8	2	1	8	1	2
9	2	2	9	1	1
10	2	2	10	1	1
11	1	1	11	1	1
12	1	2	12	2	1
13	2	2	13	1	1
14	1	1	14	1	2
15	1	1	15	1	1
16	2	2	16	1	1
17	1	1	17	1	2
18	2	2	18	1	2
19	1	1	19	1	2
20	1	2	20	2	2
21	2	2	21	2	2
22	1	1	22	1	2
23	2	1	23	1	1
24	2	2	24	2	2
25	1	1	25	1	1
26	2	2	26	2	1
27	1	1	27	2	1
28	1	1	28	1	1

- Hasil dari Skenario Uji Coba 2

1. Tabel hasil akurasi tiap *fold* yang didapatkan dengan dua *hidden layer* dan jumlah neuron [5 5].

Fold ke-	Akurasi (%) Uji Coba ke-				
	1	2	3	4	5
1	85,71	67,86	75,00	67,86	75,00
2	67,86	78,57	60,71	67,86	71,43
3	67,86	67,86	53,57	60,71	71,43
4	82,14	78,57	71,43	75,00	75,00
5	75,00	82,14	82,14	82,14	67,86
6	85,71	82,14	85,71	85,71	78,57
7	71,43	85,71	67,86	82,14	64,29
8	75,00	71,43	71,43	75,00	67,86
9	82,14	78,57	67,86	75,00	60,71
10	67,86	71,43	67,86	67,86	67,86
Rata-rata	76,07	76,43	70,36	73,93	70,00

2. Tabel hasil prediksi tiap *fold* dengan *truth class* dari rata-rata akurasi tertinggi uji coba dengan dua *hidden layer* dan jumlah neuron [5 5].

Fold 1			Fold 2		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	1	2	1	2	2
2	2	2	2	1	1
3	2	1	3	1	1
4	2	2	4	1	2
5	2	1	5	1	2
6	1	1	6	2	2
7	1	1	7	1	2
8	1	2	8	2	2
9	2	1	9	2	2
10	2	2	10	1	1
11	1	1	11	1	1
12	1	1	12	2	1
13	1	1	13	2	1
14	2	1	14	2	2
15	2	2	15	1	1
16	2	2	16	2	2
17	2	2	17	2	2
18	2	1	18	1	1
19	1	1	19	2	2
20	1	1	20	2	2
21	2	2	21	1	2
22	1	1	22	2	2
23	2	2	23	2	2
24	2	1	24	1	1
25	1	1	25	1	1
26	2	1	26	1	1
27	2	2	27	2	2
28	2	2	28	2	2

Fold 3		
Data Tes ke-	Aktual	Prediksi
1	1	1
2	2	1
3	1	1
4	1	2
5	2	2
6	1	1
7	2	2
8	2	2
9	1	2
10	1	1
11	1	1
12	2	2
13	2	2
14	2	1
15	1	2
16	1	1
17	2	2
18	1	1
19	1	2
20	1	1
21	2	1
22	2	2
23	2	1
24	1	1
25	1	2
26	2	2
27	2	2
28	1	1

Fold 4		
Data Tes ke-	Aktual	Prediksi
1	2	2
2	2	2
3	2	1
4	2	1
5	1	2
6	1	2
7	1	1
8	2	2
9	1	1
10	2	2
11	2	1
12	1	1
13	2	2
14	1	1
15	2	2
16	2	2
17	2	2
18	2	2
19	2	2
20	2	2
21	1	1
22	2	2
23	1	1
24	1	1
25	2	2
26	2	1
27	1	1
28	1	1

Fold 5			Fold 6		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	1	1	1	2	2
2	2	1	2	1	1
3	1	1	3	2	2
4	2	2	4	1	2
5	1	1	5	2	2
6	1	1	6	2	2
7	1	1	7	1	1
8	2	2	8	1	1
9	2	2	9	1	1
10	1	1	10	1	1
11	1	1	11	2	2
12	2	2	12	1	1
13	1	1	13	1	1
14	2	2	14	1	1
15	1	1	15	2	2
16	2	2	16	1	1
17	2	2	17	1	2
18	2	2	18	1	1
19	1	1	19	1	1
20	2	2	20	1	2
21	1	1	21	1	1
22	2	1	22	2	2
23	2	2	23	2	2
24	2	1	24	1	2
25	2	1	25	2	1
26	1	1	26	2	2
27	1	1	27	1	1
28	2	1	28	2	2

Fold 7			Fold 8		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	2	2	1	2	2
2	2	2	2	2	2
3	1	1	3	1	1
4	1	1	4	2	2
5	1	1	5	2	2
6	2	2	6	2	2
7	1	1	7	2	2
8	1	1	8	2	2
9	2	2	9	1	2
10	2	1	10	1	1
11	2	2	11	1	2
12	1	1	12	2	2
13	2	2	13	1	1
14	2	2	14	2	2
15	1	1	15	2	2
16	1	1	16	1	1
17	2	1	17	1	2
18	1	1	18	1	2
19	1	1	19	2	2
20	1	1	20	1	2
21	1	1	21	1	1
22	1	1	22	1	2
23	2	1	23	2	2
24	1	1	24	2	2
25	2	2	25	2	2
26	1	2	26	1	1
27	1	1	27	2	1
28	2	2	28	1	2

Fold 9			Fold 10		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	2	1	1	2	2
2	1	2	2	2	2
3	1	1	3	2	1
4	2	2	4	1	1
5	2	2	5	2	1
6	1	2	6	1	1
7	2	1	7	2	2
8	2	1	8	1	2
9	2	2	9	1	1
10	2	2	10	1	1
11	1	1	11	1	1
12	1	2	12	2	1
13	2	2	13	1	1
14	1	1	14	1	1
15	1	1	15	1	1
16	2	2	16	1	1
17	1	1	17	1	1
18	2	2	18	1	2
19	1	1	19	1	2
20	1	1	20	2	2
21	2	2	21	2	2
22	1	1	22	1	2
23	2	2	23	1	1
24	2	2	24	2	2
25	1	1	25	1	1
26	2	2	26	2	1
27	1	1	27	2	2
28	1	1	28	1	1

- Hasil dari Skenario Uji Coba 3

1. Tabel hasil akurasi tiap *fold* yang didapatkan dengan satu *hidden layer* dan *learning rate* 0,05.

Fold ke-	Akurasi (%) Uji Coba ke-				
	1	2	3	4	5
1	75,00	71,43	71,43	78,57	78,57
2	60,71	67,86	64,29	64,29	60,71
3	57,14	57,14	57,14	64,29	71,43
4	71,43	78,57	75,00	75,00	75,00
5	71,43	75,00	71,43	75,00	71,43
6	92,86	75,00	78,57	67,86	85,71
7	78,57	85,71	71,43	71,43	75,00
8	67,86	82,14	75,00	82,14	67,86
9	71,43	71,43	75,00	85,71	71,43
10	71,43	67,86	78,57	57,14	78,57
Rata-rata	71,79	73,21	71,79	72,14	73,57

2. Tabel hasil prediksi tiap *fold* dengan *truth class* dari rata-rata akurasi tertinggi uji coba dengan satu *hidden layer* dan *learning rate* 0,05.

Fold 1			Fold 2		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	1	2	1	2	1
2	2	2	2	1	1
3	2	1	3	1	1
4	2	1	4	1	2
5	2	2	5	1	1
6	1	1	6	2	2
7	1	1	7	1	2
8	1	2	8	2	2
9	2	1	9	2	2
10	2	2	10	1	1
11	1	1	11	1	1
12	1	1	12	2	1
13	1	1	13	2	2
14	2	1	14	2	1
15	2	2	15	1	2
16	2	2	16	2	2
17	2	2	17	2	1
18	2	2	18	1	1
19	1	1	19	2	2
20	1	1	20	2	2
21	2	2	21	1	2
22	1	1	22	2	2
23	2	2	23	2	2
24	2	1	24	1	1
25	1	1	25	1	2
26	2	1	26	1	1
27	2	2	27	2	2
28	2	2	28	2	2

Fold 3			Fold 4		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	1	1	1	2	2
2	2	1	2	2	2
3	1	1	3	2	1
4	1	2	4	2	1
5	2	1	5	1	2
6	1	1	6	1	2
7	2	2	7	1	1
8	2	2	8	2	2
9	1	2	9	1	1
10	1	1	10	2	2
11	1	1	11	2	1
12	2	1	12	1	1
13	2	2	13	2	2
14	2	1	14	1	1
15	1	1	15	2	2
16	1	2	16	2	2
17	2	1	17	2	2
18	1	1	18	2	2
19	1	1	19	2	2
20	1	1	20	2	2
21	2	1	21	1	1
22	2	2	22	2	2
23	2	1	23	1	2
24	1	1	24	1	1
25	1	2	25	2	2
26	2	1	26	2	2
27	2	2	27	1	1
28	1	1	28	1	1

Fold 5		
Data Tes ke-	Aktual	Prediksi
1	1	1
2	2	1
3	1	1
4	2	1
5	1	1
6	1	1
7	1	1
8	2	1
9	2	2
10	1	1
11	1	1
12	2	2
13	1	1
14	2	2
15	1	1
16	2	2
17	2	2
18	2	2
19	1	1
20	2	2
21	1	1
22	2	1
23	2	2
24	2	1
25	2	1
26	1	1
27	1	1
28	2	1

Fold 6		
Data Tes ke-	Aktual	Prediksi
1	2	2
2	1	1
3	2	2
4	1	2
5	2	2
6	2	2
7	1	1
8	1	1
9	1	1
10	1	1
11	2	2
12	1	1
13	1	1
14	1	1
15	2	2
16	1	2
17	1	2
18	1	2
19	1	1
20	1	2
21	1	1
22	2	2
23	2	2
24	1	2
25	2	1
26	2	2
27	1	1
28	2	2

Fold 7			Fold 8		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	2	2	1	2	2
2	2	2	2	2	2
3	1	1	3	1	1
4	1	1	4	2	2
5	1	1	5	2	2
6	2	2	6	2	2
7	1	1	7	2	2
8	1	1	8	2	2
9	2	2	9	1	2
10	2	1	10	1	1
11	2	2	11	1	2
12	1	2	12	2	2
13	2	2	13	1	1
14	2	2	14	2	2
15	1	1	15	2	1
16	1	2	16	1	1
17	2	2	17	1	1
18	1	1	18	1	2
19	1	1	19	2	2
20	1	1	20	1	2
21	1	1	21	1	1
22	1	1	22	1	1
23	2	1	23	2	2
24	1	1	24	2	2
25	2	2	25	2	2
26	1	1	26	1	1
27	1	1	27	2	2
28	2	2	28	1	1

Fold 9		
Data Tes ke-	Aktual	Prediksi
1	2	2
2	1	2
3	1	1
4	2	1
5	2	2
6	1	2
7	2	1
8	2	1
9	2	2
10	2	2
11	1	1
12	1	2
13	2	2
14	1	1
15	1	1
16	2	2
17	1	1
18	2	1
19	1	1
20	1	1
21	2	2
22	1	1
23	2	1
24	2	2
25	1	1
26	2	2
27	1	1
28	1	1

Fold 10		
Data Tes ke-	Aktual	Prediksi
1	2	2
2	2	2
3	2	2
4	1	2
5	2	1
6	1	1
7	2	2
8	1	2
9	1	1
10	1	1
11	1	1
12	2	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	2
19	1	2
20	2	2
21	2	2
22	1	2
23	1	1
24	2	2
25	1	2
26	2	1
27	2	2
28	1	1

- Hasil dari Skenario Uji Coba 4

1. Tabel hasil akurasi tiap *fold* yang didapatkan dengan dua *hidden layer* dan *learning rate* 0,1.

Fold ke-	Akurasi (%) Uji Coba ke-				
	1	2	3	4	5
1	82,14	75,00	85,71	67,86	85,71
2	64,29	67,86	64,29	71,43	71,43
3	60,71	60,71	75,00	67,86	67,86
4	71,43	78,57	78,57	71,43	75,00
5	78,57	78,57	78,57	75,00	64,29
6	89,29	89,29	92,86	67,86	78,57
7	78,57	71,43	71,43	67,86	71,43
8	85,71	78,57	75,00	78,57	82,14
9	75,00	60,71	71,43	75,00	78,57
10	64,29	67,86	67,86	64,29	67,86
Rata-rata	75,00	72,86	76,07	70,72	74,29

2. Tabel hasil prediksi tiap *fold* dengan *truth class* dari rata-rata akurasi tertinggi uji coba dengan satu *hidden layer* dan *learning rate* 0,1.

Fold 1			Fold 2		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	1	2	1	2	2
2	2	2	2	1	1
3	2	2	3	1	1
4	2	2	4	1	2
5	2	2	5	1	2
6	1	1	6	2	2
7	1	1	7	1	2
8	1	2	8	2	1
9	2	2	9	2	2
10	2	2	10	1	1
11	1	1	11	1	2
12	1	2	12	2	1
13	1	1	13	2	2
14	2	2	14	2	2
15	2	2	15	1	2
16	2	2	16	2	2
17	2	2	17	2	2
18	2	2	18	1	1
19	1	2	19	2	2
20	1	1	20	2	2
21	2	2	21	1	2
22	1	1	22	2	2
23	2	2	23	2	2
24	2	2	24	1	2
25	1	1	25	1	2
26	2	2	26	1	1
27	2	2	27	2	2
28	2	2	28	2	2

Fold 3			Fold 4		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	1	1	1	2	2
2	2	1	2	2	2
3	1	1	3	2	1
4	1	1	4	2	2
5	2	1	5	1	1
6	1	2	6	1	2
7	2	2	7	1	1
8	2	2	8	2	2
9	1	2	9	1	2
10	1	1	10	2	2
11	1	1	11	2	1
12	2	2	12	1	1
13	2	2	13	2	1
14	2	1	14	1	1
15	1	1	15	2	1
16	1	1	16	2	2
17	2	2	17	2	2
18	1	1	18	2	2
19	1	1	19	2	2
20	1	1	20	2	2
21	2	1	21	1	1
22	2	2	22	2	2
23	2	2	23	1	1
24	1	1	24	1	1
25	1	2	25	2	2
26	2	2	26	2	2
27	2	2	27	1	1
28	1	1	28	1	1

Fold 5		
Data Tes ke-	Aktual	Prediksi
1	1	2
2	2	1
3	1	1
4	2	1
5	1	1
6	1	1
7	1	1
8	2	2
9	2	2
10	1	1
11	1	1
12	2	2
13	1	1
14	2	2
15	1	1
16	2	2
17	2	2
18	2	2
19	1	1
20	2	2
21	1	1
22	2	1
23	2	2
24	2	2
25	2	1
26	1	1
27	1	1
28	2	1

Fold 6		
Data Tes ke-	Aktual	Prediksi
1	2	2
2	1	1
3	2	2
4	1	1
5	2	2
6	2	2
7	1	1
8	1	1
9	1	1
10	1	1
11	2	2
12	1	1
13	1	1
14	1	1
15	2	2
16	1	1
17	1	1
18	1	1
19	1	1
20	1	2
21	1	1
22	2	2
23	2	2
24	1	2
25	2	2
26	2	2
27	1	1
28	2	2

Fold 7			Fold 8		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	2	1	1	2	2
2	2	2	2	2	2
3	1	1	3	1	1
4	1	2	4	2	2
5	1	1	5	2	2
6	2	2	6	2	2
7	1	2	7	2	2
8	1	1	8	2	2
9	2	2	9	1	2
10	2	1	10	1	1
11	2	2	11	1	2
12	1	2	12	2	2
13	2	2	13	1	1
14	2	2	14	2	2
15	1	1	15	2	2
16	1	2	16	1	2
17	2	1	17	1	2
18	1	1	18	1	2
19	1	1	19	2	2
20	1	1	20	1	1
21	1	1	21	1	1
22	1	1	22	1	1
23	2	1	23	2	2
24	1	1	24	2	2
25	2	2	25	2	2
26	1	1	26	1	2
27	1	1	27	2	1
28	2	2	28	1	1

Fold 9			Fold 10		
Data Tes ke-	Aktual	Prediksi	Data Tes ke-	Aktual	Prediksi
1	2	1	1	2	2
2	1	2	2	2	2
3	1	1	3	2	2
4	2	2	4	1	1
5	2	2	5	2	1
6	1	2	6	1	1
7	2	1	7	2	2
8	2	2	8	1	2
9	2	2	9	1	1
10	2	1	10	1	1
11	1	1	11	1	1
12	1	2	12	2	1
13	2	2	13	1	1
14	1	1	14	1	1
15	1	1	15	1	1
16	2	2	16	1	1
17	1	1	17	1	1
18	2	1	18	1	2
19	1	1	19	1	2
20	1	1	20	2	1
21	2	2	21	2	2
22	1	1	22	1	2
23	2	1	23	1	1
24	2	2	24	2	2
25	1	1	25	1	1
26	2	2	26	2	1
27	1	1	27	2	1
28	1	1	28	1	1

- Hasil dari Skenario Uji Coba 5

Metode	Akurasi (%) Uji Coba ke-					Rata-Rata Akurasi (%)
	1	2	3	4	5	
Random Forest	63,93	69,29	62,86	66,07	67,5	65,93
SVM	73,57	73,57	74,29	72,50	74,64	73,71
Naïve Bayes	73,21	70,36	71,07	70,00	70,36	71,00
ANN	75,00	72,86	76,07	70,72	74,29	73,79

BIODATA PENULIS



Penulis, **Nurhamidah Tyas Palupi**, lahir di Nganjuk, 10 Oktober 1996. Penulis menempuh pendidikan sekolah dasar di SD Negeri Berbek 1, Nganjuk. Melanjutkan pendidikan sekolah menengah pertama di MTs Negeri Berbek, Nganjuk dan selanjutnya di SMA Negeri 2 Nganjuk. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya.

Selama kuliah, penulis aktif dalam organisasi tingkat jurusan maupun fakultas serta beberapa kepanitian. Diantaranya penulis berpartisipasi sebagai anggota *Departement of Internal Affair* BEM FTIf (2015-2017), anggota Departemen Dalam Negeri HMTC (2015-2016), sekretaris Departemen An-Nisa KMI (2016-2017), anggota biro Danus Schematics 2015 dan 2016.

Penulis juga mengikuti kegiatan pelatihan, diantaranya berpartisipasi sebagai peserta aktif LKMM Pra Tingkat Dasar FTIf 2014 dan peserta LKMM Tingkat Dasar FTIf 2015.

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Komputasi Cerdas Visi (KCV) dan memiliki ketertarikan dalam bidang Data Mining, Komputasi Biomedik, Pengolahan Citra Digital dan Visi Komputer. Penulis dapat dihubungi melalui email: tyastepe@gmail.com.